Commodore

Cena 12 tys. zł
nr indeksu 355275

3-4'93

KEDAM

Miesięcznik Użytkowników Komputerów C-64 i Amiga



Koło Fortuny

Modem do PACKET RADIO

Commodore



nr indeksu 355275

Wydawca:
KEBAB - sp. z o.o.
ul. Wojciechowskiego 28
PL - 71-476 Szczecin
telefon (091) 776-74

Redaguje kolegium
w składzie:
Krzysztof Kobus
Patryk Łogiewa
Grzegorz Mikuła
Krzysztof Moroń
Marcin Orłowski
Zbigniew Piotrowicz
Miłosław Smyk
Paweł Sołtysiński

Redaktor naczelny:
Patryk Łogiewa

Szef działu AMIGA:
Krzysztof Kobus
tel. (091) 525-336

Szef działu C-64:
Paweł Sołtysiński
tel. (091) 776-74

Kontakt elektroniczny:
KOBUSKPS@PLSZUS11

Redakcja nie zwraca nie zamówionych materiałów oraz zastrzega sobie prawo wprowadzania zmian w otrzymanych rękopisach.

Wydawca nie odpowiada za treść zamieszczanych ogłoszeń.

Projekt okładki:
Tomasz Kuczyński

Commodore



PRENUMERATA

Każdy egzemplarz zakupiony bezpośrednio u nas kosztuje odpowiednio:
numery: 1; 2/3; 4; 5; 6'92 - **9,5 tys. zł**
numery: 7/8; 9; 10'92 oraz następne - **11 tys. zł.**
(UWAGA! - nakład numeru 6 '92 jest wyczerpany, lecz posiadamy jeszcze pewną ilość nru 5 '92).

Oznacza to, że można zamówić numery zaległe jak też zaprenumerować jeszcze nie wydane. Odbywa się to tylko poprzez dokonanie odpowiedniej wpłaty na nasze konto. Na odwrocie każdego odcinka kuponu wpłaty należy dokładnie napisać, których numerów wpłata dotyczy.

W przypadku prenumeraty można zamawiać numery do końca aktualnego okresu "małej prenumeraty". W roku 1993 "mała prenumerata" będzie obejmować cztery egzemplarze, z podziałem roku na trzy okresy:
I - numery 1, 2, 3, 4 - 4 egz. x 11 tys. = 44 tys. zł
II - numery 5, 6, 7, 8 - 4 egz. x 11 tys. = 44 tys. zł itd.
(jeśli do tego czasu cena nie ulegnie zmianie).

Prosimy nie przysyłać do redakcji dowodów wpłat.

Nasze konto:
Pomorski Bank Kredytowy
II Oddział w Szczecinie
numer konta: 368113-25771-136

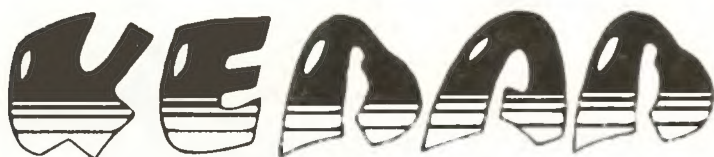
Podajcie dokładny adres, imię i nazwisko zamawiającego oraz numery egzemplarzy, których wpłata dotyczy na odwrocie każdego z odcinków blankietu wpłaty.

I ostatnia prośba - **wszelką korespondencję, wpłaty etc. kierujcie tylko i wyłącznie na adres redakcji.**

REKLAMA

Ogłoszenia drobne od osób indywidualnych (do 10 słów na kuponie wyciętym z III-ciej strony okładki) przyjmujemy bezpłatnie. Ogłoszenia drobne od osób prawnych oraz zawierające powyżej 10 słów są płatne w całości po 1000 zł za słowo.
Ogłoszenia ramkowe (minimalny format 20 cm^2):
1 cm^2 - 4,5 tys. zł, cała strona 2,5 mln. zł,
cała IV strona okładki - 4 mln. zł, 1/2 tej strony - 2,5 mln. zł,
dodatkowy kolor - odpowiednio 50 % drożej.

Ogłoszenia płatne prosimy przesyłać listem poleconym.



Nr 3-4 marzec-kwiecień 1993

Od redakcji

Witajcie! Spotykamy się po raz trzeci w tym roku, choć zbliża się już koniec kwietnia. Niestety przeżywaliśmy (znowu) zimowo-wiosenne kłopoty natury organizacyjno-technicznej, czego efektem jest opóźnienie oraz to czego Wy (i my również) tak nie lubicie - numer łączony. Na początek o Ankiecie z Nru 9/92. Informacje, które uzyskaliśmy dzięki niej, pozwoliły nam dokładniej poznać Wasze oczekiwania względem Kebaba i choć zapewne nie będziemy w stanie zadowolić wszystkich, dołożymy wszelkich starań aby uczynić to najpełniej. Zgodnie z obietnicą każdy uczestnik Ankiety został wpisany do naszego redakcyjnego archiwum, natomiast wymienione poniżej osoby wylosowały bezpłatną premumeratę Kebaba do końca 1993 roku. Są to:

- Andrzej Pappelbaum z Grudziądza
- Dominik Szczap z Biskupic Ołobocznych
- Grzegorz Bogucki z Pruszcza Gdańskiego
- Sebastian Raszkiewicz z Bartoszyc
- Robert Nowak z Czarnkowa
- Łukasz Jarczyk z Bochni
- Mariusz Juszczak z Bielawy
- Krzysztof Oszmiańczuk z Żar
- Jan Mucha z Goleniowa
- Damian Rozenblut z Żor

Dodatkowo w ramach wdzięczności dla tych spośród Czytelników, którzy zadali sobie trud wypełnienia (i wysłania) ankiety rozlosowaliśmy dwie nagrody - niespodzianki. Posiadaczem najnowszej wersji oryginalnego cartridge'a Action Replay do C-64 został **Michał Cajsel** z Łodzi, a dla Amigi - **Adam Nowak** z Katowic. Jednocześnie chcelibyśmy zaprosić Was po raz kolejny do wspólnego redagowania naszego pisma; piszcie jeśli macie jakieś pomysły, problemy, propozycje czy uwagi - jest to jedyna droga, aby Kebab stał się naprawdę Waszym pismem. Jak już pewnie zauważyliście zwiększyliśmy objętość do 48 stron, zmniejszyliśmy czcionkę, zmianie uległa szata graficzna, pojawią się już wkrótce nowe działy, m.in. konkursy; postaramy się też ożywić nasze czarno-białe strony.

Niestety wobec wszechobecnych podwyżek, które również i nas dotknęły, nie uda nam się utrzymać ceny pisma na dotychczasowym poziomie i od numeru 5'93 będzie ona wynosić 15 tys. zł za egzemplarz. Wiemy, że jest to duży wydatek dla Waszych kieszeni, lecz pozwoli to nam na uczynienie naszego pisma bardziej interesującym i obszerniejszym.

Redakcja

Spis treści :

02 Z kraju i ze świata
03 Commodore
04 Arexx bez Arexx'a?
05 Demo - Przewodnik
08 Koło Fortuny dla C-64
08 List do redakcji
10 PC Emulator V1.63
12 Ze Sceny
14 Cyfrowy pejzaż
15 AUTOSTARTY na C-64
16 O wektorach jeszcze raz
19 ELYSIUM
20 Budowa modemu do pracy Packet Radio
23 Zapomniane urządzenia - cz.2
24 Asembler na C-64 - odc. kolejny
26 Mapa pamięci Amigi
28 JAK ZROBIĆ WŁASNY TURBOLOADER - czyliprogramowanie stacji 1541/71
32 AMOS VII
34 O drukarkach kolorowych słów kilka
37 FORUM
38 "Ancient Battles" czyli starożytne bitwy
41 Gry na C-64
43 Cheaty i wskazówki do gier
44 Ogłoszenia drobne
45 Listingi: Scroller - w uzupełnieniu do nru 2'93 KOŁO FORTUNY LOADER 1 i 2
48 Kebab Public Domain 3 na Amigę
Dr Boczek na III stronie okładki

COMMODORE

NOWOŚCI

Firma Parth TGalen oferuje nowe biblioteki matematyczne. Jak obiecują producenci wszelkie funkcje w nich zawarte wykonywane mają być do 180 razy szybciej od standardowych funkcji matematycznych, zawartych w bibliotekach IEEE podwójnej precyzji, dostarczanych wraz z systemem operacyjnym przez Commodore. Jeśli rzeczywiście dzięki zastosowaniu optymalnych algorytmów liczenia, funkcje te będą pracować zgodnie z reklamowaną prędkością, to zwykła pięćsetka będzie w stanie liczyć z wydajnością porównywalną z Amigą 3000 zaopatrzoną w koprocesor 68882. O tym, jakie korzyści niesie za sobą użycie tychże bibliotek wraz z programami do Ray-Traceingu, czy obróbki obrazu nie trzeba chyba nikogo przekonywać. Oczywiście biblioteki te nie wymagają żadnych rozszerzeń sprzętowych i są w stanie współpracować z wszelkimi programami wykorzystującymi standardowe biblioteki matematyczne.

Firma Supra za jedyne 200 dolarów oferuje posiadaczom Amigi 500 "dopalacz" Supra-Turbo28. Owo urządzenie wielkości rozszerzenia pamięci SupraRam500RX (opisywanego przez nas w Kebabie 11-12'92) podłączane z lewej strony obudowy zaopatrzone jest w mikroprocesor 68000 taktowany częstotliwością 28MHz. Testy przeprowadzone na programach: ADPro, Lharc oraz VistaPro wykazują około trzykrotne przyspieszenie w stosunku do standardowej Amigi. Przystawka zaopatrzona jest dodatkowo w zewnętrzny przełącznik umożliwiający pracę w standardowym trybie, w celu zapewnienia kompatybilności z niektórymi grami i starszymi użytkami.

Classic Cocepts - firma specjalizująca się w produkcji na potrzeby video wysokiej jakości czcionek, lansuje pakiet EuroFonts Video 3.0 zawierający 14 krojów, każdy w 4 rozmiarach, międzynarodowych alfabetów. Poza literami węgierskimi, albańskimi, rumuńskimi, krajów skandynawskich znajdują się tam również litery polskie. Fonty są kompatybilne z oprogramowaniem dostarczanym wraz z popularną w USA kartą VideoToaster. Cena pakietu: $129.

Amerykański lider w produkcji rozszerzeń sprzętowych do Amigi, a ostatnio również oprogramowania, firma GVP zaprezentowała dwie karty do A1200. Pierwsza z nich oznakowana A1230 TURBO+, zaopatrzona jest w procesor 68EC030 taktowany 40MHz, 1MB RAM z możliwością rozszerzenia do 32MB oraz podstawkę pod opcjonalny koprocesor matematyczny. Karta instalowana jest do gniazda znajdującego sie pod spodem A1200, co pozwala zachować warunki gwarancji podczas jej montowania. Drugi produkt oznakowany A1200 SCSI/RAM+ to szybki kontroler SCSI dysku twardego, którego brak w podstawowej konfiguracj spowodował ogólną krytykę Commodore. Dodatkowo na płycie znajduje się 1MB pamięci z możliwością rozszerzenia do 8MB, oraz podstawka pod opcjonalny koprocesor matematyczny.

Doskonały program do morphingu o nazwie Morph Plus doczekał się już aktualizacji. Wersja 1.2 posiada sporo nowych ciekawostek. A to: możliwość bezpośredniego generowania zbiorów w formacie IFF-ANIM5, możliwość ustawienia rozmaitych algorytmów umożliwiającą wykonywanie morphingu z różną jakością (i w różnym czasie). Oprócz tego pakiet został jeszcze bardziej dopasowany do maszyn wyposażonych w AGA.

Kolejną ciekawostką na łączach C-64 Amiga jest fakt zaprezentowania przez firmę Faustdick Computer, interface'u umożliwiającego podłączenie do Amigi... magnetofonu Datasette od C-64. Nie byłoby może w tym nic rewelacyjnego gdyby chodziło tylko o wykorzystanie tego interface'u na potrzeby programowego emulatora C-64 o nazwie "The A64 Package" znanego między innymi z dysków Freda Fisha. Tak! Dzięki zastosowaniu tego urządzonka możemy nie tylko ładować programy z C-64 do emulatora ale również możemy wykorzystać nasz wierny Datasette jako streamer do twardego dysku Amigi. Producent dołącza odpowiednie oprogramowanie i zapewnia, że szybkość transmisji takiego streamera wynosi... bardzo dużo!

Jak już mowa o Fish'u to należałoby powiedzieć o pogłosce, która ostatnio krąży po rozmaitych sieciach komputerowych. Otóż mowa w niej, że Fred Fish zamierza zrezygnować z dalszego prowadzenia serii dysków PD na Amigę. Zmusza go ponoć do tego znaczny spadek liczby prenumeratorów serii.

Rośnie ostatnio konkurencja pomiędzy dwiema najbardziej uznanymi kartami graficznymi do Amigi. Zarówno Impact Vision 24 jak i Opal Vision doczekały się nowych wersji oprogramowania. Dołączany do karty IV-24 program graficzny "Macro Paint" został znacznie zmodyfikowany, gdyż w swojej pierwszej wersji "nie dorastal do pięt" produktowi firmy Centaur Development odpowiedzialnej za produkcję i dystrybucję programu "Opal Paint". Ten ostatni z kolei umożliwia aktualnie "hotlinkowe" połączenie z legendarnym już ADPro, tak aby wszystko czego nie można zrobić w Opal'u mogło być wykonane w ADPro i na odwrót. Oby tak dalej!

Na zakończenie mamy zaszczyt (chyba jako pierwsi) poinformować naszych Czytelników o zbliżającym się: ELYSIUM & ANTIA DEMO-PARTY. Party ma się odbyć w dniach 28-30 czerwca w hali sportowo-widowiskowej Pałacu Młodzieży w Tarnowie. Więcej na ten temat znajdziecie w następnym KEBAB'ie.

Ze względu na brak czasu, nie byliśmy w stanie osobiście odwiedzić targów CEBIT 93 w Hannowerze, niemniej jednak osoba blisko związana z redakcją była na miejscu i postanowiła przekazać swoje spostrzeżenia czytelnikom KEBAB'a.

Moje wspomnienia z Cebitu.

Do Hanoweru pojechałem obejrzeć największe targi komputerowe. Rozmach organizatorów, doskonała organizacja mimo bardzo zmiennej pogody i zmęczenia podróżą zachęcały do zwiedzania. Wystawiły się firmy z całego świata. Nie było żadnych super nowości, gorących hitów. Oczywiście popędziłem zaraz do pawilonu pierwszego, gdzie wystawiała się firma Commodore... Prawdę mówiąc nie spodziewałem się takiego rozmachu z jakim Commodore prezentowało swoje wyroby. Były: Nieśmiertelny Commodore 64 z dyskiem twardym, Amiga 500, Amiga 600, Amiga 1200, Amiga 3000T (Tower), CDTV i oczywiście A4000. Może niewielu z Was wie, że Commodore jest również producentem doskonałych (jak na PC) "smutniaków" czyli PC-tów. Commodore pokazało swoją serię 386SX-25, 386-33C, 486SX-25, 486-33C, T486-50C, T486DX-66C i notebooka 486SX-25LTC - oczywiście "ALL IN ONE". Commodore wyraźnie zmienia swe oblicze. Po raz pierwszy od niepamiętnych czasów pokazało swoje wyroby w sposób nie pozostawiający wiele do życzenia. Każdy komputer miał swoje stanowisko z opisem np."A4000-ANIMATION" lub "T486-50C-WINDOWS" i opiekuna (operatora). Zwiedzający mieli dokładne rozeznanie czego mogą oczekiwać. Były pokazy filmowe z podkładem muzycznym ukazujące możliwości animacyjne i graficzne Amigi 4000. Została stworzona ciepła atmosfera wśród zwiedzających.

Na stoiskach prezentowano również kartę "PICASSO" firmy VILLAGE-TRONIC za 598DM- autokonfigurującą się, do Amigi2000/3000/4000, z pamięcią 1MB, max rozdzielczością 1280*1024 w 16 mln. kolorów (w trybie True-Color).

Norweska firma SCALA prezentowała swój nowy produkt: SCALA MULTIMEDIA. GVP prezentowało programy "Cinemorph" i "ImageFX" oraz 24-bitową kartę graficzną z funkcją PICTURE-IN-PICTURE i flicker fixerem (prawdopodobnie iV-24 przyp.red.).

Kolejnym produktem który zaprezentowano na targach był "DIRECTOR II" firmy TRI-TEC MARKETING. Jest to 32 bitowa maszyna graficzna z 4MB VRAM i 4MB CPU DRAM, z procesorem TMS34020. Jej parametry to: max rozdzielczość 1024*1024, pixele w 32 bitach przy częstotliwości ramki 32 MHz, genlock- pasmo 6 MHz, auto-locking z wejściem RGB. Prezentowano również karty: RAINBOW VIDEO-LAYER, RAINBOW III,"dopalacze" TURBO- HARD-CARD - firmy GVP, oraz programy graficzne: DMI-WRITE,PHOBOS V.39,TRUE-PAINT (grafika 24 bitowa); muzyczne : GRAFIK-SEQUENCER, AMIGA MIDI, GRAFIK PLAN, MINGNON LAS, SOUND-2-MIDI KONVETERTER ,MIDI-TOOL-DISK, oraz programy AMIGA-ORBIT, AMIGA-MENSCH,MEDIALINK I REAL 3D U.2.

Na inne stoiska zostało mi już niewiele czasu, lecz nie żałowałem. Całą powrotną drogę w marzeniach widziałem się przed moją A2000 z wspaniałymi: dopalaczem i kartą 24 bitową. Ach...

WILK

Commodore

Tyran czy protektor Amigi?

Polityka rynkowa Commodore'a od zarania komputerów budziła kontrowersje, stając się obiektem krytyki środowisk, którym zależało bardziej na popularyzacji produkowanych przez niego modeli, niż na napełnianiu kieszeni szefom firmy. Postarajmy się prześledzić kolejno wydarzenia ostatnich lat, aby móc wyrobić sobie zdanie na powyższy temat.

Wszystko zaczęło się w roku 1984 -firma Amiga stała na granicy przepaści, zmuszona była sprzedać swój projekt nowego supermikrokomputera. W szranki stają dwie wrogie sobie korporacje: Atari i Commodore. Obydwie upatrują w projekcie nowego modelu między innymi szansę zemszczenia się na przeciwniku.

Przez zachłanność Jack'a Tramiela, do podpisania umowy nie dochodzi, a firma Amiga zostaje wykupiona przez Commodore'a. Tramiel zaczyna kompletować zespół do skonstruowania Wunderwaffe - jak się później okazało - modelu 520 ST. Tymczasem Commodore przystępuje do prac nad ucieleśnieniem dopiero co zakupionego projektu. Niebieski gigant IBM wprowadza PC AT z procesorem 80286, a w Stanach Zjednoczonych w firmie Apple rodzi się MacIntosh.

W roku 1985 Atari wypuszcza na rynek pierwszy model z serii ST. Z marszu zdobywa sobie popularność, o czym najlepiej świadczy fakt sprzedaży w roku następnym 40 tysięcy komputerów ST w samej tylko RFN. O Amidze jeszcze nikt nie słyszy, lecz jak się później okazało, jest to cisza przed burzą. Czas pracuje na korzyść IBM, szczególnie ze względu na upowszechnianie się pecetów i stałe rozwijanie się produkcji z Dalekiego Wschodu.

Wkrótce nadchodzi wreszcie pamiętna chwila - z hukiem rodzi się Amiga. Specjalistyczna prasa komputerowa zafascynowana jest niespotykanymi dotąd w urządzeniach tego typu możliwościami graficznymi i dźwiękowymi, które są na dokładkę okraszeniem multitaskingu. System o tak szokujących parametrach jednak nie upowszechnia się. Na przeszkodzie staje bariera cenowa i brak oprogramowania. Wielu wskazuje na kulejący marketing Commodore'a i w wielu przypadkach jest to prawdą.

Jeżeli chce się wprowadzić nowy standard, to operacja taka musi być koniecznie poprzedzona szeroko zakrojoną kampanią reklamową i promocyjną, wymagane są także duże inwestycje na skłonienie firm software'owych do zainteresowania się oferowanym sprzętem. Nic by nie zaszkodziło Commodore'owi, by przez pewien przejściowy okres zajął się również tym drugim, niesprzętowym aspektem swego modelu, przykładowo poprzez zlecenie wielu firmom produkcji komercyjnych programów, a nie tylko Microsoftowi, który nie popisał się dołączanym przy zakupie Amigi Basic'iem.

Przyczyny nieudolnej operatywności na rynku sięgają swymi korzeniami roku 1984, gdy wspomniany już wcześniej Jack Tramiel odchodzi z Commodore'a ciągnąc za sobą inżynierów, oraz cały dział marketingu. Osłabiony Commodore potrzebuje czasu na dobór nowych i sprawnych ludzi, przyjęcie skoordynowanej polityki, oraz na zwykłe otrząśnięcie się po całym zamieszaniu z Tramielem. Ponadto nie znamy wszystkich faktów związanych z pojawieniem się Amigi, a w tym najważniejszego: sytuacji finansowej firmy w tym okresie. Być może składanie do kupy tegoż komputera na tyle zaabsorbowało finanse, że pozostało tylko wydać go na świat i w nadziei czekać czy przeżyje.

Rok 1986 to czas walki Amigi o rynek, oraz data pojawienia się nowego peceta opracowanego w firmie Compaq, wyposażonego w procesor 80386. Natomiast Atari milczy.

Rok 1987 okazuje się przełomowym dla Amigi: na rynek wprowadzone zostają modele 500 i 2000 (starszej Amidze dla odróżnienia zostaje dodany przydomek 1000). Poprawione są niedociągnięcia poprzedniego modelu, stabilizuje się powoli rozsądniejsza cena. Uznanie zdobywa dobra decyzja firmy o dwutorowości produkcji: na rynek komputerów domowych (A-500) i profesjonalnych (A-2000).

Również jednak w tym okresie, Commodore'owi zarzuca się przede wszystkim brak uderzenia w serce bezpośredniego konkurenta, czyli w serię ST. Wyposażenie nowych komputerów w interfejs MIDI, z pewnością by pozbawiło przeciwnika jego ostatniego atutu, pozwalając tym samym Amigom na zdominowanie amatorskiego rynku muzycznego. Choć MIDI można oczywiście osobno dokupić, zawsze to jednak dodatkowy problem.

To co zapewniło modelom Atari przetrwanie to właśnie interfejs muzyczny, sprytna polityka cenowa, oraz udane wejście na rynek DTP z modelami Mega ST (mającymi swą premierę w tym czasie) i doskonałym programem DTP (Desktop Publishing) o nazwie Calamus. W firmie IBM początek dwuletniego przestoju w dziedzinie rozwoju komputerów "kompatybilnych", między innymi związanego z próbą wprowadzenia linii PS/2.

W roku 1988 światło dzienne ogląda po raz pierwszy Amiga

2500. Właściwie nie wiadomo o co Commodore'owi chodzi: A-2500 nie oferuje żadnych nowości technicznych, gdyż A-2000 z akceleratorem (procesor Motorola 68030, koprocesor 68882) i innymi kartami (np. Flicker-Fixer) to prawie to samo. Nie ma przecież sensu lansowanie "nowego" produktu, który jest niczym innym jak modelem poprzednim z powkładanymi kartami rozszerzającymi, tym bardziej, że takie działanie zniechęca producentów alternatywnych peryferii.

Amiga 2500 UX to niezdecydowany i koślawy krok w kierunku systemu operacyjnego Unix. Gwoździem do trumny obu modeli była ich cena, której ze względu na bardziej wrażliwych czytelników, nie będę podawał. Jak widać zamysł o kryptonimie 2500 to posunięcie chybione, a całe szczęście tylko w tym, że żadna z konkurencyjnych firm nie zaprezentowała w tym roku niczego godnego uwagi.

Rok 1989 - oj działo się, działo... To chyba udany rok dla wszystkich. Atari zmniejsza dysproporcje w stosunku do Amig, wypuszczając nowe modele STE i Mega STE.

Ukazuje się duma fanów pecetów, czyli procesor Intel 80486. Nie trzeba długo czekać, aby nie zaczęto ich instalować do PC. Niewątpliwe sukcesy obu firm. Jednak istnieje między Atari a IBM'em podstawowa różnica: o ile Atari (i Commodore również) stale udoskonala swe maszyny, to ewolucja pecetów po wprowadzeniu do powszechnego użytku procesora i486 się zakończyła.

Do dnia dzisiejszego, firmy nie stać na nic bardziej heroicznego, a oprócz kosmetycznych poprawek i kart graficznych nie pojawiło się w dziedzinie hardware'u zbyt wiele rzeczy godnych odnotowania. W sztabie Commodore również odświętna atmosfera. Promowanie przez komputerowy świat technik multimedialnych jako kierunku rozwoju, szczęśliwie zbiega się z wysiłkami nad opracowaniem Amigi CDTV (Commodore Dynamic Total Vision). Amiga CDTV ukazuje się rok później, lecz mimo braku poważniejszej konkurencji do prymatu w tym obszarze, dość trudno zdobywa sobie uznanie i popularność.

Główną przyczyną jest to, że przeciętny człowiek nie dorósł jeszcze do tego typu udogodnień.

CDTV jest przeznaczona przecież dla zwykłych ludzi, jako swego rodzaju sprzęt z serii artykułów gospodarstwa domowego, ułatwiającego i uprzyjemniającego życie.

Drugą, dość istotną sprawą jest fakt, że występują jeszcze (jak na razie) pewne problemy z wykonywaniem pirackich kopii oprogramowania na dyskach typu CD-ROM. Jednak jeżeli Commodore nie zaniecha tego kierunku i będzie go stale pielęgnować, to pewnego dnia odpowiedni czas dla tego typu urządzeń nadejdzie i będziemy się zastanawiali "Jak kiedyś można było żyć bez telewizora, sprzętu Hi-Fi i CDTV?".

C.D.N.

Zbigniew "Zybul" Plotrowicz

Arexx bez Arexx'a ?

Czy można korzystać z zalet Arexx'a nie posiadając Arexx'a ? Okazuje się że tak! Wystarczy posiadać Cshell 5.17. Program ten potrafi bowiem obsługiwać porty Arexx'a równie dobrze jak sam Arexx!



Możemy zatem pisać interface'y miedzy programami używając scriptów Cshell'a. Do obsługi portów służą jedynie dwa rozkazy:

```
rxrec i rxsend
```

Pierwszy z nich służy do uaktywnienia portu wbudowanego w Cshell. Można zatem sterować jego pracą z zewnątrz (z innego programu). Składnia rozkazu jest następująca:

```
rxrec [<portname>]
```

Nazwa portu jest opcjonalna (standardowo rexx_csh). Po wykonaniu tego rozkazu shell zawiesza się i oczekuje na rozkazy z zewnątrz. Jedyną metodą zakończenia tego stanu jest przesłanie mu komendy

```
bye
```

Drugi rozkaz służy do sterowania innych programów posiadających port Arexx'a. Oto jego składnia:

```
rxsend [-lr] portname
command..command
```

Parametry -l i -r są opcjonalne. Portname to nazwa portu (np rexx_ced albo TxEd1), command to przesyłany rozkaz. Przykładowo sekwencją:

```
rxsend rexx_ced cedtof-
ront open test.c jmp to
line 20
```

prześle do portu Cygnus Edytora rozkazy załadowania pliku test.c i skoczenia do 20 linii tekstu.

Inna metoda przetestowania tej możliwości to uruchomienie dwóch Cshell'i w osobnych oknach.

W jednym z nich wpisujemy:

```
rxrec shell_1
```

zaś w drugim:

```
rxsend shell_1 "dir df0:"
```

co powoduje wyświetlenie przez pierwszy shell katalogu df0:. Wypróbuj inne rozkazy! Komunikacje należy zakończyć sekwencją:

```
rxsend shell_1 "bye"
```

co spowoduje zakończenie pracy pierwszego shell'a w trybie odbierania rozkazów z portu.

Dla zainteresowanych dodaję, że opcja `-r` powoduje ustawienie zmiennej `_result` na zwracaną przez program wartość (spróbuj przesłać do Ced'a rozkaz status 32 !) zaś opcja `-l` przesyła całą linię jako JEDNĄ komendę.

K.M.

Demo-Przewodnik



Od początku istnienia, Amidze towarzyszyły programy, których jedynym zadaniem było zaprezentowanie oglądającemu wspaniałych możliwości graficznych i dźwiękowych tej maszyny. Pierwsze demo (bo o nich tu mowa) napisane zostało przez samych twórców tego komputera, aby zachęcić firmy komputerowe do inwestowania w Amigę.

To co nastąpiło potem przeszło zapewne najśmielsze oczekiwania sceptyków. Nastąpił rozwój sceny, masowo zaczęły pojawiać się rozmaite grupy, które zajęły się pisaniem programów, w tym także programów demonstracyjnych, szokujących zwykłych zjadaczy chleba rozmaitymi, niesamowitymi efektami, pozornie niemożliwymi do osiągnięcia. Powstawały (i powstają nadal) coraz to nowe dema podnoszące coraz wyżej poprzeczkę ustawianą przez poszczególnych programistów, wnosząc przy tym do informatycznego słownika coraz to nowe pojęcia określające poszczególne efekty. Aby pomóc Czytelnikom zorientować się w tym wszystkim, przygotowaliśmy małe omówienie poszczególnych, najczęściej spotykanych w demach efektów.

Zacznijmy od najpopularniejszego i zarazem najprostszego do wykonania "efektu" czyli oczywiście od

SCROLL'a.

Napisanie scroll'a czyli przesuwającego się po ekranie tekstu, jest przez niektórych programistów traktowane niemal na równi z chrzestem równikowym czy też jako odpowiednik pasowania na rycerza i w 95 przypadkach na 100 jest to pierwsza poważniejsza procedura większej części Amigowych programistów. Z biegiem czasu ten, początkowo prosty, efekt ulegał stopniowym modyfikacjom. Litery zaczęły poruszać się po różnych wymyślnych torach (np. po kręgu) czy też wyginały się na różnych krzywych, a że jako pierwszą użytą krzywą była sinusoida nazwano to po prostu

SINUS-SCROLL'em.

Algorytm (a właściwie procedura go realizująca) jest na tyle czasochłonna iż przez dłuższy czas w demach mogliśmy oglądać co najwyżej dwa jednoczesne sinus-scroll'e. Jednak od pewnego czasu, za sprawą odmiennej techniki wykonanania (która jest bardzo szybka ale i pamięciożerna) ich liczba drastycznie wzrosła, zaś ostatnim rekordem jest (a może było) szesnaście sinus-scroll'i!

Następnym stopniem wtajemniczenia w karierze większości koderów (czyli programistów piszących głównie w assemblerze) jest, równie popularna, trójwymiarowa grafika wektorowa zwana często "wektorówką". Idea jest tak stara jak ... podręczniki geometrii, albowiem wszelkie niezbędne wzory na której wektorówka bazuje, większość z Was powinna poznać już w szkole podstawowej.

Pierwsze figury wektorowe (nieśmiertelny sześcian), które mieliśmy okazję podziwiać w starszych demach (lub obecnie w produkcjach początkujących grup), to tzw.

LINE VECTORS

czyli wektory liniowe (lub jak kto woli: druciaki), gdzie poszczególne ściany sprawiają wrażenie przezroczystych, albowiem na ekranie rysowane są ich wszystkie możliwe krawędzie. Daje to dość często, szczególnie w przypadku sześcianu, wizualny efekt rozjeżdżania się poszczególnych ścian lub odwrotnej perspektywy. Zapobiec temu można poprzez wyeliminowanie niewidocznych ścian, czyli tych które, patrząc z punktu widzenia obserwatora, leżą "z tyłu" danej figury. Tak liczone, i rysowanie druciaki, z eliminacją niewidocznych ścian, określane są często mianem

HIDDEN-LINE VECTOR.

Następnym krokiem naprzód były wektory wypełniane:

FILLED VECTORS,

w których każda ściana wypełniona była danym kolorem. Należy przy tym zauważyć, że z punktu widzenia programisty, różnica pomiędzy wektorówką wypełnianą a "normalną" polega w głównej mierze na sposobie rysowania danej figury (używany jest do tego specjalny tryb Blitter'a) i oczywiście jej wypełnianiu (także Blitter !), przy czym wyliczanie zasłoniętych ścian jest w tym momencie zalecane.

Dlaczego tylko zalecane?

Otóż rozpatrzmy przypadek figury, której wszystkie ściany leżą w jednej płaszczyźnie. Zatem jasno widać, że nie istnieje możliwość zasłonięcia jakiejkolwiek za ścian przez pozostałe. Taką przypadek określa się mianem

PLANE VECTORS

(ang. plane - płaski). A jeśliby tak poszczególne figury miały kształty liter, to odpowiednio je animując moglibyśmy uzyskać...

VECTOR SCROLL!

Nie trzeba chyba nikogo przekonywać, że jest to znacznie bardziej efektowne niż zwyczajny scroll. Proszę przy tym zauważyć, że odpowiednie połączenie kilku starych, "oklepanych" technik prowadzi do narodzin nowych i ciekawszych efektów (a oto przecież w demach chodzi).

Gdy ludzi przestały już cieszyć dema, w których główną atrakcją był latający po ekranie sześcian, zaczęto zastanawiać się nad uatrakcyjnieniem wektorówki. Zaczęły pojawiać się różne fikuśne obiekty, w których dodatkową atrakcją były np. ich ruchome elementy, lub których ściany miast jednolitych były wypełnione konkretnym wzorkiem - tzw.

STENCIL VECTORS,

czy też nawet całe miasta zbudowane na wektorach. Uważny czytelnik zauważył zapewne, że musiano przy tym rozwiązać jeden dodatkowy problem, a mianowicie częściowe przesłanianie się poszczególnych ścian. Dla odróżnienia od normalnej, konwencjonalnej wektorówki, nosi ona nazwę

INCONVEX VECTORS (czyli wektory niekonwencjonalne).

Wróćmy na chwilkę do problemu widoczności ścian. Ich obliczanie pozwala nam skrócić czas potrzebny na narysowanie bryły, bowiem odpadają od razu ściany, które docelowo i tak byłyby niewidoczne. Wynika to z prostego założenia, że wszystkie ściany danego obiektu nie są przezroczyste.

Niektórzy koderzy zaczęli się swego czasu nad tym zastanawiać: a gdyby jednak rysować wszystkie ściany, ale tylko w taki sposób, aby te które teoretycznie są niewidoczne (czyli są z tyłu figury) były przyciemnione, to wtedy, przy odpowiednim ustawieniu palety kolorów i odpowiednim rysowaniu, powinno to dawać wrażenie pół-przezroczystości figury. I rzeczywiście daje, zaś w słowniczku fanów dem pojawiło się kolejne nowe określenie -

Programy Demonstracyjne

GLENZ VECTOR

(zainteresowanym techniczną stroną tego efektu proponuję rozrysować układ kombinacji pixeli na, przykładowo, 3 bitplane'ach oraz odpowiadające im numery kolorów).

Aby podnieść atrakcyjność dem, zaczęto przy grafice wektorowej symulować źródło światła, które oświetlało poszczególne ściany szalejącej po ekranie figury. Efekt ten dał się szeroko poznać jako

LIGHTSOURCED VECTORS.

Wprawdzie początkowo źródło światła było nieruchome (co znacznie upraszcza obliczenia) ale z czasem i z tym się uporano. Następną innowacją która zaczęła pojawiać się w kolejnych produktach różnych grup były

RUBBER VECTORS,

w których obiekt sprawia wrażenie jakby był wykonany z masy plastycznej, co w fazie końcowej daje wrażenie "przelewania" lub "ciągnięcia" się ścian. Zauważmy, że jeśli zdefiniujemy poszczególne obiekty jako litery, to otrzymamy, podobnie jak w przypadku Vector Scroll'a, nowy rodzaj scrollingu. Z powodów natury głównie technicznej, litery w takim scroll'u obraca się wokół jednej tylko osi. Stąd też miast rubber scroll (jak zapewne niektórzy z Was sądzili) efekt ten nosi nazwę

TWIST SCROLL'a.

Jeśli myślicie, że to już koniec atrakcji opartych na geometrii przestrzennej to się niestety mocno mylicie, jednak na krótką chwilę zostawimy to w spokoju, bowiem nie z samymi wektorami człowiek żyje... Dużą rolę w każdym programie demonstracyjnym pełni także muzyka i grafika. Jednak jako że w stosunku do muzyki wizualnych efektów nie wymyślono zbyt wiele, pozostając głównie przy wskaźnikach wysokości dźwięku zwanych

EQUALIZER-ami

(a których wykonanie bazuje w głównej mierze na opisanych powyżej technikach) więc rozdział ten pominiemy (aczkolwiek nic nie stoi na przeszkodzie by i tutaj ktoś z Was wprowadził coś nowego - swego czasu szalenie

podobały mi się equalizery wykonane w formie... sedesów, gdzie poziom dźwięku ukazywały ruszające się w rytm muzyki klapy!).

A grafika... tu wszystko zależy od uzdolnień odpowiedzialnego za nią człowieka, jednak zawsze poprzez np. specjalne jej wyświetlenie można zwiększyć wrażenie jaki dany rysunek ma wywrzeć na widzu. Do takich ciekawych efektów należy przykładowo

FLOOD

w którym poszczególne linie "spływają" na ekran układając się w rysunek. Oczywiście tak wyświetlony obrazek można dalej obrabiać poprzez np. jego różne deformacje czy też płynne powiększanie (STRETCH/ZOOM czyli rozciąganie/powiększanie) wzdłuż osi X, Y lub obu jednocześnie tak, że docelowo każdy pixel rysunku może osiągnąć nawet wielkość całego ekranu!

Inną metodą efektownego pokazania grafiki, metodą zdobywającą coraz większą popularność, jest

MORPHING.

Terminem tym określa się płynne przechodzenie jednego obrazu w drugi, bazując przy tym na poszczególnych pixelach obrazu (ci, którzy widzieli teledysk Black or White wiedzą już o czym mówię). Nie jest to nieosiągalne, albowiem dla Amigi wypuszczono już kilka programów które umożliwiają właśnie taką obróbkę obrazu. Ci, którzy mieli możliwość ich używania, dobrze wiedzą że takowa operacja do najszybszych nie należy, co nie zmienia faktu iż w kilku demach pojawiły się już namiastki morphingu (oczywiście pomijamy tu przeliczone wcześniej, np. za pomocą CineMorph'a czy DPaint'a, animacje).

Siłą rzeczy (a właściwie szybkości procesora) ograniczają się one do niewielkich, monochromatycznych obrazków choć i to może wkrótce ulec zmianie dzięki pojawieniu się Amigi 4000.

Wracajmy do naszych wektorów... Powiedzieliśmy już wcześniej, że nowe efekty powstają zwykle przez mariaż kilku innych, wcześniej już znanych. Spróbujmy postąpić podobnie i tym razem. Dobrze, zapytają niektórzy, lecz co może łączyć rysunek ze zwykłą wektorówką? Pozornie oba te elementy nie mają ze sobą nic wspólnego. Ale mogą mieć!

Proponuję sięgnąć po Kebaba 10/92 do artykułu "Mapa pamięci". Wśród opisu poszczególnych rejestrów Blitter'a znajdziecie rejestr odpowiadający za wzór kreślonej linii. Jeśli ktoś jeszcze nie wie do czego zmierzam, to proponuje poeksperymentować z zamieszczonym tam programem, zmieniając za każdym razem wartość rejestru BLTBDAT.

Pewien sprytny programista wpadł zatem na pomysł aby jako wzór do rysowania linii umieszczać fragmenty rysunków (albowiem, jak to w komputerach bywa, także rysunek zapamiętywany jest w formie bitów, bajtów itd.), co pozwoliłoby nam na np. obracanie rysunków: wystarczyłoby tylko policzyć początek i koniec każdej linii, jako wzór podstawić rusunek i rysować, rysować, rysować. Niestety rzeczywistość jest bardziej okrutna. Przez kilka ograniczeń nałożonych przez konstruktorów na Blitter, pisząc procedury

PHOTO VECTOR'ów

zmuszeni jesteśmy wpisywać do BLTBDAT kolejne fragmenty rysunku co każde narysowane 16 pixeli, co niestety drastycznie wpływa na prędkość takich operacji.

Opisane powyżej efekty można zaliczyć obecnie do wiodących aczkolwiek nie jedynych. Nie należy zapominać o pozostałych, równie ciekawych, choć rzadziej spotykanych w demach efektach. Zacznijmy od rozmaitych

BOB'ów.

Terminem tym (dobrze znanym miłośnikom Amosa) określa się dowolne obiekty poruszające się po ekranie. Mogą to być rozmaite kuleczki, kwadraciki czy też kółka, które mogą poruszać się w dowolnym celu po dowolnych krzywych. Nic nie stoi na przeszkodzie aby symulować ruch takich obiektów w przestrzeni, stosując analogiczne wzory do stosowanych przy grafice wektorowej czy też stosując wzory na sferę (najprościej dla współrzędnych biegunowych).

Także składając kilkakrotnie jedno przekształcenie można osiągnąć bardzo ciekawe wyniki. Przykładem niech będzie

TV-BOX,

czyli wektorowy sześcian, na którego ściankach obracają się inne obiekty wektorowe. Także scroll może stać się interesującym efektem jeśli np. owiniemy go na kuli lub puścimy go po wymyślnej funkcji przestrzennej lub też tak zanimujemy litery, aby sprawiały wrażenie płynących po falującej wodzie (WATER SCROLL).

Różnych możliwości są tysiące i zależą w głównej mierze od pomysłowości koderów. Kto wie, okazać się może, że któryś z Czytelników (Czytelniczek?!) wypuszczając swoje demo pośle wszystkie opisane tutaj efekty do lamusa. Aby nie być dłużej gołosłownym odsyłam zainteresowanych do dem, w których wyżej opisane efekty można obejrzeć "na żywo":

Alpha & Omega/Pure Metal Coders,
Can't be/Vision,
Deformations/Deform,
Elysium/Sanity,
Enigma/Phenomena,
Hardwired/Silents-Crionics,
Ice/Silents,
Joyride/Phenomena,
Little Vector Preview/Complex,
Static Chaos/Silents,
Turmoil/Sanity.

Marcin Orłowski

PS: Zainteresowanch technicznymi aspektami niektórych z opisanych wyżej efektów odsyłam do wcześniejszych numerów Kebaba (6/92, 10/92).

SOFTWARE

Koło Fortuny 64

Ani się obejrzeliśmy, jak popularność telewizyjnej zabawy przerosła najśmielsze oczekiwania. Tłumy Polaków zasiadają przed telewizorami i emocjonują się szczęściem, domyślnością i dużymi sumami pieniędzy.

Zasada zabawy jest prosta: należy zgadywać przygotowane przez organizatorów hasła, opierając się tylko na kilku dostępnych literach. Na dobrą sprawę przypomina to trochę rozwiązywanie krzyżówek.

Telewizyjna zabawa doczekała się nawet w Polsce własnych komputerowych wersji. Jak do tej pory widziałem wersje na PC, Amigę i C64. Wszystkie one mają jedną wadę - raczej trudno byłoby umieścić ich listingi na łamach jakiejkolwiek gazety a poza tym w wersji nas najbardziej interesującej - na C64 - nie można było ustalać własnych haseł. Chcąc niechcąc, aby uczynić możliwym tą publikację - musiałem trochę nagiąć fakty dla naszych potrzeb:

- zrezygnowałem z "koła" jako takiego; nie ma losowania ilości pieniędzy (no bo jakich?) i co za tym idzie - brak w grze pól "Bankruta" i "Straty kolejki";

- w grze nie ma ograniczenia w używaniu samogłosek.

Jest za to możliwość (i o to głównie chodziło) własnoręcznego przygotowywania haseł (a nawet całych ich zestawów).

Program należy wpisać do komputera wykorzystując Korektor lub najzwyklejszy debugger (należy wtedy pomijać kody kontrolne, które umieszczone są w nawiasach na każdym końcu linii). Przed uruchomieniem należy nagrać wynik swojego wklepywania na dysk lub taśmę.

Po prawidłowym wykonaniu powyższych operacji możemy uruchomić program poprzez RUN. Naszym oczom powinny ukazać się na ekranie opcje programu:

1. Edycja hasła i start gry
2. Wczytanie haseł i start gry
3. Edycja haseł

Opcja pierwsza (1) pozwoli nam na grę tylko z jednym hasłem - po jej wybraniu edytujemy hasło na ekranie, ustalamy jego rodzaj (powiedzenie, osoba, cytat itp.) i możemy wołać osoby, które nie znając hasła będą się świetnie bawiły. Każdy z biorących udział graczy jest posiada numer kolejny (od 1 do 3) a obok odpowiadających im rubryk znajdują się dwa parametry wskazujące na jakość ich zabawy - liczba określająca ilość odgadniętych prawidłowo liter i liczba odgadniętych całych haseł.

Jak już wspomniano powyżej, nie ma ograniczeń w używaniu samogłosek - nie trzeba ich kupować itd. Jedyną przestrzeganą zasadą jest możliwość odgadywania hasła po uprzednim odgadnięciu przynajmniej jednej litery w tej kolejce. Aby dokonać próby odgadnięcia całego hasła wystarczy je wpisać; o ile podasz więcej niż jedną literę - komputer zostaje przez to poinformowany, iż nie chodzi Ci o odgadnięcie kolejnej litery lecz o odgadnięcie całego hasła. Zrezygnować z odgadywania aktualnego hasła można poprzez wciśnięcie klawisza RUN/STOP.

Opcja druga (2) pozwala na wgranie z taśmy lub dyskietki zestawu uprzednio przygotowanych haseł i rozpoczęcie gry. W ten sposób można się na przykład wymienić z kolegą zrobionymi przez siebie zestawami i dobrze się bawić.

Opcja ostatnia, trzecia (3) pozwala na przygotowanie zestawu haseł. Definiujemy je jedno po drugim wraz z podaniem komputerowi typów tych haseł. Po każdorazowej definicji następuje pytanie komputera, czy chcesz kontynuować edycję. Jeżeli nie będziesz dłużej edycją zainteresowany - program zapyta o nazwę i nagra dane na dysku lub taśmie.

Miłej zabawy życzy

Paweł Sołtysiński

OD REDAKCJI

!!! SZANOWNA REDAKCJO !!!

Lubię wasze pismo gdyż jest na luzie. Nie zasmucacie czytelników suchymi tekstami, lecz wprowadzacie w nie nieco humoru. Niestety piszę właśnie ze względu na humor. Większa jego część opiera się o komputery ATARI ST i IBM PC. Jestem posiadaczem PROTECH-a AT-286, 16MHz, 1 MB RAM-u, dysk twardy 40 MB, kartę graficzną TVGA, monitor kolorowy, joystick, mysz, drukarkę STAR LC24-10 - słowem mówiąc niezła konfiguracja. Jak do tej pory nie zdarzyło mi się aby ktoś wyśmiewał się z mojej maszyny lub nazwał ją "smutasem". Z reguły wszyscy chwalili znakomitą grafikę (SPACE QUEST 4, KINGS QUEST 5 i 6, POLICE QUEST 3, WING COMMANDER 2) i narzekali na słabiuteńką muzykę. Podłączałem wtedy COVOXA i puszczałem moduły z AMIGI. Odnosiło to skutek, gdyż słuchający zazwyczaj byli zadowoleni z tego co usłyszeli. Mimo tego nie staram się nikomu wmówić, że AMIGA lub ATARI ST to złom. Wprost przeciwnie, podziwiam je za dobrą muzykę w konfiguracji podstawowej i sporą ilość dem, które uwielbiam. Co robicie natomiast Wy?

Za wszelką cenę chcecie w waszym piśmie pokazać wyższość AMIGI nad innymi komputerami (ATARI ST, IBM PC). Wydaje mi się że jesteście przez to trochę nie lubiani przez posiadaczy tych komputerów, którzy czytają wasze pismo. Czy nie moglibyście przestać drukować głupawych tekstów i obrazków typu tych w KEBABIE 1'93 - str. 9 i

12? Wydaje mi się że nie zauważacie takiego czegoś jak SVGA, SOUND BLASTER, WINDOWS 3.1, MS-DOS 5.0, CORELL DRAW 3.0... Zresztą jeżeli uważacie AMIGĘ za taką samodzielną, to po co buduje się emulatory "smutasa" czy ATARI ST?

W miarę możliwości proszę o odpowiedź na łamach KEBABA na nurtujące mnie pytania. Czekając na wyjaśnienia kontynuuję czytanie.

Bartosz Górecki (BLUE)

Drogi Czytelniku,

Twój list (może to zabrzmi dziwnie) sprawił nam ogromną radość. Świadczy on bowiem o tym, że istnieją jeszcze na świecie ludzie, którzy potrafią (choćby na moment) uchylić z oczu klapki z napisem "mam peceta" i spojrzeć nieco szerzej na świat. Niestety w zdecydowanej większości przypadków klapki są nieusuwalne. I ten właśnie fakt, a nie sam "pecet" jest przede wszystkim obiektem naszej ironii.

O ile nic nie mamy przeciwko urządzeniom "pecekompatibilnym" dopóki znajdują się na swoim miejscu tzn. w biurze i służą do prostego przetwarzania tekstu lub obliczeń, o tyle strasznie smutnym jest fakt, że istnieją już ludzie, którzy kupują tego typu sprzęt mając na myśli zastosowania typu DTV (Desktop Video) czy (pożal się Boże) ray-tracing, animację itp. W zupełności nie interesuje ich fakt, że do tego typu zastosowań istnieją już długo rozwiązania zarówno sprzętowe jak i programowe, pozwalające użytkownikowi Amigi na znacznie wyższy stopień kreatywności.

Jako żywo przypomina mi się przypadkowo zasłyszana niedawno rozmowa dwóch "profesjonalistów", z którymi dane mi było odbyć kilkugodzinną podróż pociągiem. Otóż po około 40-minutowym rozwodzeniu się nad zaletami systemu 486DX przeszli oni do zachwalania dokupionej właśnie do niego 24-o bitowej karty graficznej. Konkluzją kolejnych czterdziestu minut było stwierdzenie, że karta ta jest jednak prawdziwie profesjonalnym sprzętem gdyż istnieje nawet możliwość wyświetlenia na jej ekranie fontów zainstalowanych w Windows'ach... W tym momencie nie wytrzymałem i parsknąłem śmiechem na cały przedział. Ten sam profesjonalista kilka dni wcześniej rozmawiał z moim kolegą i zadał mu pytanie: Cóż to za nowy komputer pan sobie kupił panie Krzysiu...? Na odpowiedź: Amigę, zawołał: Coo??? Amigę to mają moje dzieci! Oczywiście nie dał sobie wytłumaczyć, że jest to Amiga 4000 wyposażona w 18MB RAM'u, kartę Opal Vision a wkrótce również VIVID 24. Na koniec należy jeszcze dodać iż zarówno "Pan Krzysio" jak i "profesjonalista" zajmują się tym samym tylko pochodzą z dwóch konkurencyjnych firm. Różnica polega na tym, że pierwszy z nich postanowił sprawdzić jaki sprzęt będzie się najlepiej nadawał do tego co zamierza robić i posiadał wcześniej zarówno "peceta" jak i Macintosh'a. W tym momencie dokonał ŚWIADOMEGO wyboru.
Drugi natomiast stwierdził, że skoro komputer to znaczy PC, więc nie ma innych możliwości. Czy nie zgodzisz się ze mną Drogi Czytelniku, że jest to bardzo SMUTNY fakt? Nie pozostaje jednak nic innego jak tylko się śmiać i wprowadzać trochę radości do smutnego świata. Ja osobiście posiadam również dzięki świadomemu wyborowi Amigę 4000 i nie chciałbym nigdy abym zamiast ikonek Workbench'a lub ulubionego DirOpusa musiał oglądać PONOWNIE napis "Norton Commander...". Byłby to z pewnością jeden ze smutniejszych dni w moim życiu...
Wróćmy jednak na chwilę do Twojego listu. Czy nie zauważamy produktów wymienianych przez Ciebie? Mam nadzieję, że dzięki kilkukrotnemu wspominaniu o świadomym wyborze sprzętu dałem do zrozumienia, że wszystkie one są nam znane. Aby jednak nie być gołosłownym zróbmy małą wyliczankę.

SVGA - to w końcu coś o nieco większych możliwościach graficznych od premierowej Amigi 1000 (z roku 1985). Oczywiście należy zapomnieć o szybkości obsługi ekranu bo tu należałoby porównywać Amigę 1000 (z roku 1985) do PC486DX-50+ MHz (z roku 1992).

SOUND BLASTER - to wreszcie coś o zbliżonych możliwościach dźwiękowych do premierowej Amigi 1000 (z roku 1985).

WINDOWS 3.1 - to "w pełni profesjonalna" nakładka graficzna na -> MS-DOS, która niestety do dzisiaj nie posiada niektórych możliwości Workbench'a z premierowej Amigi 1000 (z roku 1985). Mam tu na myśli przede wsystkim true-multitasking. Co do objętości twardego dysku jaką zajmuje po zainstalowaniu, oraz szybkości działania nie będę się wypowiadał gdyż każdy "pecetowiec" te sprawy zna...

MSDOS 5.0 - to "profesjonalny" system operacyjny pozwalający m.in. na: formatowanie dyskietek, kopiowanie zbiorów, wczytywanie i uruchamianie programów jak również "w pełni profesjonalnych" nakładek typu -> WINDOWS. Nie wiemy niestety tylko co jest w nim takiego niesamowitego i czym przewyższa on inne systemy tego typu. Na pewno natomiast nie pozwala na zastosowanie nazw zbiorów typu np. "MojPierwszyProgram" co w Amiga-DOS'ie i w systemach na niektórych innych komputerach jest najzupełniej normalne.

COREL DRAW 3.0 - to bardzo dobry program do tworzenia grafiki opartej o obiekty wektorowe (linie, krzywe Bezier itp.) Bogactwo możliwości stawia go w czołówce programów tego typu niezależnie od rodzaju komputera. Wadą natomiast jest szybkość (tzn. jej brak) pracy, co w połączeniu z faktem, że pracuje on tylko pod kontrolą "w pełni profesjonalnej nakładki" daje na 386-33 prędkość porównywalną do prędkości odpowiednich programów na A500.

I jeszcze odpowiedź na ostatnie pytanie: emulatory IBM'a pisze się po to aby pokazać, że to co na zabawkowej Amidze jest już możliwe od kilku ładnych lat, na profesjonalnym pececie prawdopodobnie nie będzie jeszcze możliwe za kilka ładnych lat.

PS.

Nie jest prawdą że większa część naszego humoru opiera się o komputery innego typu. Znacznie więcej miejsca poświęcamy niekompetencji oraz indolencji redaktorów bardziej lub mniej profesjonalnych czasopism komputerowych z naszego podwórka (patrz np. rubryka "Dylematy doktora Boczka") - ani IBM ani ATARI ST nie posiadają u nas swojej rubryki, niewykluczone jednak, że utworzymy taką jeśli zainteresowanie Czytelników tym tematem będzie dostatecznie duże. Czekamy na listy w tej sprawie - piszcie na adres redakcji.

PC-Emulator V1.63

SOFTWARE

czyli jak załamać iBM'owca

"Konflikty zbrojne" we wszystkich tzw. "świętych wojnach" pomiędzy komputerami, to na szczęście tylko dyskusje między kolegami, posiadającymi inny sprzęt. Sam ten fakt jest rzeczą szalenie pozytywną, gdyż niejako zmusza najczęściej młodych ludzi do poznania swego komputera, w celu zyskania argumentów do dyskusji.

Wojna między użytkownikami Amigi i iBM'a jest już niemal przysłowiowa. Amigerów denerwuje fakt, iż Amiga mimo lepszych możliwości i rozwiązań sprzętowych nie usunęła raz na zawsze pecetów, natomiast iBM'owcy są oburzeni na myśl, że taka "zabawkowa" Amiga ma czelność równać się z "jedynie profesjonalnymi" komputerami. Do jednych nie dociera fakt, iż IBM to już po prostu standard (ale nie we wszystkich dziedzinach), a zmiana standardu pociąga za sobą ogromne koszty, stając się po prostu nieopłacalna. Drudzy uważają, że Amiga to tylko A-500 z 0,5 Mb Ram, jedną stacją dysków i pełną gamą gier. Ponadto całkowicie ignorują fakt, że jeżeli przy porównaniu walorów technicznych, bardzo mizernie wypada PC 386 w stosunku do A-3000, to 486 w stosunku do A-4000 nie wypada już wcale.

"Konflikty zbrojne" we wszyskich tzw. "świętych wojnach" pomiędzy komputerami, to na szczęście tylko dyskusje między kolegami, posiadającymi inny sprzęt. Sam ten fakt jest rzeczą szalenie pozytywną, gdyż niejako zmusza najczęściej młodych ludzi do poznania swego komputera, w celu zyskania argumentów do dyskusji. Nie chcąc zostać na uboczu tych spraw, zamieszczamy przepis na ofensywę przeciw fanom IBM'a. Aby nie była to walka jednostronna, zapraszamy pisma IBM'owskie do repliki i kontrofensywy. Z góry jednak zastrzegamy, że nikogo nie mamy najmniejszego zamiaru obrażać i tego samego wymagamy od przeciwników. Chcemy, aby ta ZABAWA była traktowana z humorem i nie wywoływała żadnych nieprzyjemności.

Wracając do tytułowego tematu... Sposobów na załamanie iBM'owca jest wiele. Dzielą się one na tanie i drogie. Do tych drogich zalicza się kupno twardego dysku i rozszerzenia pamięci (ew. Amigi wyposażonej weń od razu) i porównywanie systemów przy zbliżonych konfiguracjach. W tym numerze zajmiemy się jednak tańszym sposobem.

Obmyślając strategię naszej wojny, wyszliśmy z założenia, iż najefektywniejsze będzie pokazanie przeciwnikowi, że Amiga potrafi być nie tylko Amigą ze wszystkimi jej zaletami, ale i w pełni będzie emulować komputer oponenta. Już widzę jak rzedną miny Amigowców: "Eeee... Czy taki Transformer lub PC-Task na zwykłej Amidze 500 bez dopalacza będzie mógł konkurować z pecetem?". Odpowiedź na to pytanie jest oczywiście przecząca. Ale jest nowy proszek... przepraszam - program, który swą szybkością działania, od ręki "załatwia" XT, a w stosunku do AT 286 jest konkurencyjny. Sprawa jest więc rozwiązana:



iBM'owca załamujemy pokazując mu po prostu program o skromnej nazwie: PC-Emulator wersja 1.63 (lub nowsza jeżeli się ukaże). Program ten jest typu Public Domain i jest dostępny między innymi na Kebab Public Domain Dysku nr 3. Emulator ten został napisany specjalnie z myślą o naszej wojnie. Osobiście wypróbowałem go niedawno na pewnym IBM'owcu, przez co na własne oczy zobaczyłem, jaka to skuteczna broń. Ten właśnie fakt bezpośrednio skłonił mnie do napisania niniejszego artykułu.

Zalety PC Emulator'a:

1. Szybkość.

Oto komunikat wygenerowany programem Speed PC Info :

Test 1: IBM PC XT: 1.0s,
IBM PC AT: 0.43s,
IBM PC 386: 0.26s,
Your PC 0.67s,

Test 2: IBM PC XT: 2.2s,
IBM PC AT: 1.01s,
IBM PC 386: 0.43s,
Your PC 1.91s,

Test 3: IBM PC XT: 2.9s,
IBM PC AT: 2.22s,



cie jego demonstracyjna wersja, gdyż zamieszczenie oryginalnego byłoby naruszeniem praw autorskich), które mówiąc w języku slangowym "chodzą jak burza".

Dysk ten podzielony jest na trzy części (oczywiście nie fizycznie!). Na jednych ścieżkach (track'ach) nagrane jest intro, inne dostępne są normalnie dla systemu AmigaDos, zaś pozostałe ścieżki są zarezerwowane na programy pod kontrolą Ms-Dos'a (nie należy więc ingerować w strukturę dysku).

Oprócz wspomnianego Norton Commandera, mamy możliwość przetestować programy takie jak: Modplayer do odgrywania muzyki, Memtest testujący pamięć, Viruschk do wykrywania wirusów, oraz wiele innych.

IBM PC 386: 1.39s,
Your PC 2.82s,
Test 4: IBM PC XT: 7.8s,
IBM PC AT: 3.56s,
IBM PC 386: 1.67s,
Your PC 4.35s.

Test wykonany na Amidze z procesorem 68000. Jak wyniki przedstawiają się w przypadku u-

```
sysinfo
SI-System Information, Advanced Edition, (C) Copr 1987, Peter Norton

     Operating System:  DOS 3.30
       Main Processor:  NEC V20                    Serial Ports:  3F8,2F8
         Co-Processor:  None                     Parallel Ports:  3BC,378
Video Display Adapter:  Color/Graphics
   Current Video Mode:  Text, 80 x 25 Color
Available Disk Drives:  3, A: - C

DOS reports 640 K-bytes of memory
   50 K-bytes used by DOS and resident programs
  590 K-bytes available for application programs
A search for active memory finds
  640 K-bytes main memory     (at hex: 0000-A000)
   47 K-bytes display memory  (at hex: A040-AC00)
   16 K-bytes display memory  (at hex: B800-BC00)

 Computing Index (CI), relative to IBM XT: 6.9
      Disk Index (DI), relative to IBM XT: Not computed. No drive specified.

Performance Index (PI), relative to IBM XT: Not computed.

A:\>
```

ruchomienia emulatora na Amidze 4000 z procesorem 68040, pokazuje zdjęcie. Osiągi uzyskane w czasie testu nie wymagają komentarza.

2. Kompatybilność.

Na dysku z emulatorem dołączonych jest kilka programów, m.in Norton Commander (oczywiś-

3. Niekomercyjność.

Dzięki temu każdy Amigowiec może poznęcać się nad "pececiarzem", bez żadnych nakładów finansowych i z czystym sumieniem (czy można znęcać się z czystym sumieniem?).

4. W wyniku drobnych zmian w systemie operacyjnym, emulowanemu pecetowi wydaje się, że ma zainstalowaną kartę Covox. Karta (?) Covox na prawdziwym komputerze iBM pozwala na odgrywanie modułów z Amigi. W naszym przypadku nie musimy starać się o żadne rozwiązania sprzętowe, gdyż emulator jako Covox'a będzie używał odpowiednich, specjalizowanych układów z naszego komputera. Jakość odtwarzanej w ten "okrężny" sposób muzyki, nie będzie idealnie wierna swe-



mu amigowskiemu odpowiednikowi - przecież to zawsze tylko emulacja.

Z pewnością niejeden z testujących demonstracyjną wersję PC-Emulatora, zadawał sobie pytanie, dlaczego dotychczas nikt w tak efektywny sposób nie przyspieszył emulacji,a niektórzy nawet oferowali takie "żółwie" jak Transformer. Autorzy emulatora zastosowali kilka niestandardowych, trików, a w pliku doc uchylili rąbka tajemnicy.

Cóż się zatem okazało? Piętą achillesową wszystkich emulatorów są procedury obsługi ekranu. W PC-Emulatorze przyspieszenie uzyskano poprzez stablicowanie przeliczonych adresów tyczących się ekranu i stały nadzór nad ilością wymaganych bitplane'ów. Na uwagę zasługuje szczególnie to drugie rozwiązanie. Jak wiemy, do uzyskania 16 kolorów na ekranie, potrzebne są 4 bitplane'y. Zdarza się, iż w czasie pracy z emulatorem, na ekranie są tylko dwa kolory (na przykład w trakcie wpisywania komend w pecetowskim CLI). W takiej sytuacji procedura obsługi ekranu wyłącza 3 zbędne bitplane'y, co niesamowicie przyspiesza operacje na grafice, bez żadnych różnic w wyświetlanym obrazie. Sprytne, co? Dalsze wyjaśnienia tyczyły się konwersji trybów adresowania, o których nie będę pisał, gdyż problemy te nie są mi znane. Odsyłam do pliku DOC. Niestety, te i pozostałe zabiegi przyspieszające na tyle absorbują pamięć, że posiadacze Amig z 0,5 Mb Ram, mogą od razu zapomnieć o uruchamianiu PC-Emulatora.

Na zakończenie chciałbym gorąco polecić ten sposób walki. Przyniesie on z pewnością dużo śmiechu nad zdziwionymi minami oponentów. Czekamy teraz na odpowiedź wrogiego obozu, być może to oni będą się następnym razem z nas śmiali...

Dariusz "Dr. Boczek" Boczyński

ZE SCENY

Ponownie witamy wszystkich zainteresowanych, w tej rozwijającej się coraz prężniej (oczywiście dzięki Wam) rubryce. Na marginesie muszę dodać, iż Wasz odzew na hasło “Na scenę” przeszedł nasze najśmielsze oczekiwania i co miesiąc zasypywani jesteśmy kolejnymi listami zawierającymi kolejne owoce Waszej pracy, które oczywiście będziemy sukcesywnie przedstawiać w kolejnych numerach Kebaba.

Tak się złożyło, że produkcje które mam dziś zamiar opisać z definicji należą do grupy (bynajmniej w kategorii Koła Fortuny) programów demonstracyjnych. Jestem pewien, że w “amatorskich” kręgach, tego pojęcia nie trzeba nikomu wyjaśniać (pamiętam jak dziś reakcję jednego z moich profesjonalnych wykładowców, który gdy dowiedział się, że to co aktualnie było mu dane oglądać (grafika, muzyka - i to wszystko jednocześnie?!) nazywa się programem demonstracyjnym, zapytał: “A co on demonstruje?”. Jako, że nie był to TurboPacal x.0 ani jedynie profesjonalny Windows, przeprosił i odszedł pogrążony w zadumie).

Na początek weźmy demo grupy **ABDABS** o nazwie *VEKTOR DEMO I* (pisownia oryginalna). Jest to program zajmujący cały dysk i jak sama nazwa wskazuje, oglądając możemy spodziewać się rozmaitych obiektów wektorowych. Tak też jest w rzeczywistości. Autor skoncentrował się na przedstawieniu wszechstronnego wykorzystania grafiki wektorowej poczynając od klasycznych już “druciaków” (czyli wektorówki nie wypełnionej) poprzez vector-bobs'y a kończąc na obiektach wypełnianych, które to podziwiać możemy przez większą część dema.

Demu towarzyszy interesujący podkład muzyczny, niestety, jako że ABDABS podobnie jak większość młodszych grup, cierpi na brak artystycznych dusz, wykonany przez kogoś spoza grupy. Podobnie ma się rzecz z grafiką, której lwią część wykonać musiał sam programista.

VEKTOR DEMO nie jest jedyną produkcją nadesłaną do nas przez tą grupę. Otrzymaliśmy także drugi dysk, zawierający dwa dema plikowe tego samego autora, demonstrujące poprawione, jak twierdzi on sam, procedury grafiki wektorowej oraz krótką animację. Jedyne zastrzeżenie jakie mam do nadesłanych prac to fakt użycia przez autora symbolu swastyki, której widokiem mamy wątpliwą przyjemność cieszyć się w programie VEKTOR DEMO.

I choć w załączonym liście autor tłumaczy, że w tym przypadku miał inne intencje, to jednak historia na tyle wypaczyła znaczenie tego znaku, aby większości ludzi kojarzył się zupełnie z czymś innym niż japońską symboliką.

A oto “lista płac”:

Program: Yark
Grafika: Yark





oraz adres kontaktowy:

Jarosław Mądrala
ul.Kwiatowa 33
32-540 Trzebinia

Następne demo to program autorstwa **CHARON'a** z grupy... No właśnie.

Wprawdzie w demie widnieją logosy DEFCOM ONE, lecz w przysłanym do redakcji liście możemy przeczytać, że są one pozostałością z prób współpracy z tą właśnie grupą (zaś innej nazwy nie udało nam się znaleźć). Demo nosi podwójną nazwę: AGGRESSION (widniejącą na tytułowej planszy) vel AGGRESSIA (widniejącą w jego pozostałych miejscach).

Podobnie jak w poprzednio opisywanym demie, także i tu autor duży nacisk położył na wyeksponowanie grafiki wektorowej, której





szerokie zastosowanie możemy obejrzeć na ekranie np: rozmaite wypełnianie obiekty (w tym dość ciekawie animowany sześcian) czy 9 niezależnie poruszanych, "drucianych" ostrosłupów. Nie poprzestając na tym, autor rozszerzył nieco listę pokazywanych efektów o między innymi, interesujący scroller czy prostą, aczkolwiek efektowną metamorfozę. Brak muzyka, podobnie jak w przypadku poprzedniej grupy, zmusił do użycia jako podkładu, istniejącego już modułu muzycznego (w tym przypadku jest to muzyka z gry UTOPIA), o czym, w obu przypadkach, autorzy uczciwie informują.

Do powstania opisywanego programu przyczynili się:

Kodowanie: Charon
Grafika: Pinhead, Charon

Ponieważ członkowie tej grupy deklarują chęć wstąpienia w szeregi istniejącej, większej polskiej grupy, więc zainteresowanym ewentualną współpracą podajemy adres:

Jarosław Wetoszka
ul. Pokoju 1
21-500 Biała Podlaska

Ostatnim z omawianych dziś programów będzie demo-dysk grupy **Black Bit** z Katowic. Piszę demo-dysk ponieważ na przysłanym do nas nośniku znajdujemy trzy niezależne programy: *Space Vectors, Spheres* oraz *Art Gallery*.

Dwa pierwsze to sensu scricte dema, natomiast Art Gallery to wystawa prac grupowgo grafika.

Zatem po kolei.

Space Vectors to kolejna realizacja znanego już pomysłu na grafikę 3D czyli trójwymiarową (nie mylić z grafiką wektorową). Aby uzystać wizualny efekt przestrzenności animowanych obiektów, należy oglądać całość poprzez specjalne okulary, w których jedno szkiełko ma zabarwienie niebieskie, drugie zaś czerwone, co daje wrażenie wystawania np: sześcianu z monitora.

Ci, którzy oglądali w kinie ostatnią część "przygód" Freddy Krueggera już sobie pewnie przypominają (niestety autorzy zapomnieli dołączyć owych okularów do przesyłki i musieliśmy posiłkować się pożyczonymi). Spheres to trzy poruszające się niezależnie sfery, w podstawowych kolorach, które to nakładając się na siebie powodują powstawanie mieszanych kolorów, co każdy powinien pamiętać ze szkoły podstawowej (o ile nie spał na plastyce!).

No i na koniec Art Gallery czyli mała galeria sztuki malarskiej, zawierająca osiem rysunków o różnorodniej tematyce, i jest to jak na razie pierwszy slide-show przysłany do redakcji. Zawarte w nim rysunki, to w 3/4 dzieło Axel'a D. zaś pozostałe są autorstwa Dr.And oraz Dark'a.



Mnie osobiście najbardziej spodobały się prace pierwszego z wymienionych (czyli właśnie Axel'a D.), które stoją na naprawdę dobrym poziomie (czego niestety, nie oddają zamieszczone obok fotografie, głównie z racji ograniczeń kolorystycznych).

Zobaczymy czy moje zdanie podzielają także inni, ponieważ grupa Black Bit przygotowuje się do najbliższego Demo-Party, które odbyć się w Żywcu (więcej informacji na temat tego party zamieścimy... jeśli tylko nadeślą je nam organizatorzy).

A oto autorzy:

Programowanie: Dark
Grafika: Axel D., Dr.And, Dark
Muzyka: Axel D.

oraz adres grupowego swappera:

Stanley
ul.Teodora Kulika 11
40-424 Katowice

Słowo podsumowania. Ponieważ wszystkie nadesłane dema stoją na równym poziomie, tym trudniej o jakąś szczegółową ocenę. Myślę, że każdy z autorów doskonale zna swoje możliwości oraz wie najlepiej jakie błędy popełnił, zaś nadesłane prace nie zawierały na tyle rażących błędów, żeby na ich temat się rozwodzić... Prócz jednego. Prawie żaden z opisanych programów nie chciał pracować na A500 z rozszeżeniem FAST-RAM ani też na A1200 pomimo wyłączenia pamięci Cache oraz odpowiedniego ustawienia układów graficznych.

Jedyną reakcją na taki stan rzeczy było radosne mruganie diody Power. Jest to rażący błąd, który należy z całą stanowczością wytknąć.

Prosimy testujcie swoje programy przed wysłaniem ich do nas, bowiem jest fizyczną niemożliwością byśmy do każdego nadesłanego programu szukali komputera na którym raczyłby on się uruchomić (wszak jest to wbrew zdrowemu rozsądkowi). Amiga to już nie tylko A500, lecz również A600, A3000, A1200 czy A4000! Pamiętajcie o tym gdy będziecie pisać swoje kolejne programy.

Marcin Orłowski

P.S. Prosimy także o nadsyłanie prac na sprawdzonych, najlepiej firmowych, dyskach (nadesłane do redakcji nośniki ZWRACAMY!), bowiem już kilka razy zdarzyło się nam, że sfatygowany NoName zrezygnował ze współpracy w momencie testowania danego programu!

Jak na razie udawało nam się wyjść zwycięsko z takich potyczek z opornymi dyskami i uratować program, lecz nie jest to przecież regułą!

Każdy, kto zetknął się z grafiką komputerową na pewno słyszał o fraktalach. Choć na pozór mogłoby się to wydawać niemożliwe, te abstrakcyjne, matematycznie tworzone obrazy można zaprzęgnąć do wiernego symulowania otaczającej nas rzeczywistości

CYFROWY PEJZAŻ

SOFTWARE

Można tego dokonać z pomocą Amigi i programu *Scenery Animator 2.00* stworzonego przez amerykańską firmę *Natural Graphics*. Aby przekonać się co można zrobić z jego pomocą potrzeba jednak czegoś więcej niż zwykła "pięćsetka" - niezbędne jest minimum 1.5 MB pamięci, a praca bez dysku twardego i 32 bitowego procesora nie należy do szybkich i wygodnych.

Pierwszą czynnością, którą musimy wykonać aby mieć nad czym pracować jest stworzenie mapy terenu - na szczęście nie musimy jej ręcznie rysować - wyręczą nas w tym właśnie fraktale. W tym celu klikamy na "**Fractl**", typ fraktala ustawiamy na "V 1.0", "**Seed**" na 19445, a "**Height**", czyli maksymalną wysokość na 5000.

Aby zdefiniować ostateczny wygląd terenu korzystamy z funkcji "Land". Umożliwia nam ona ustawienie następujących opcji:

"**Snow**" (śnieg) - pozostawiamy wyłączony,
"**Rock**" (skała) - wpisujemy poziom 1700,
"**Veg**" (roślinność) - poziom 1200,
"**Soil**" (gleba) - włączamy.

Parametry te można oczywiście dowolnie modyfikować, pamiętając jednak, że nie można użyć wszystkich jednocześnie z powodu ograniczonej palety 32 barw. Problem ten łatwo jest ominąć korzystając z trybu IFF24, ale o tym na koniec. Warto posadzić trochę drzew za pomocą "**Tree**" - do wyboru mamy sekwoje ("**Redwood**") lub dęby ("**Oak**"). Możemy dowolnie ustalić zakres wysokości na jakich się pojawią i ich ilość.

Za pomocą funkcji "**Water**" włączyć możemy ocean znajdujący się na określonym poziomie, na którym nawet pojawić się mogą fale ("**Waves**"), lecz nam jest on na razie niepotrzebny.

Musimy także odpowiednio ustawić oświetlenie, do czego służy opcja "**Light**". Nasze "słońce" należy umieścić w pobliżu środka, tak aby uniknąć nadmiernych cieni.

Funkcją "**Sky**" zmienić możemy także wygląd nieba - obniżyć pułap chmur, lub zwiększyć ich gęstość.

W celu wyznaczenia toru, po jakim poruszać się będzie obserwator wybieramy "**Map**", co powoduje wyświetlenie mapy terenu.

Animacja w Scenery Animator jest rozwiązana w dość wygodny sposób, za pomocą "klatek kluczowych" (ang. key frames).

Oznacza to, że na mapie zaznaczamy punkty, przez które przechodzić ma tor ruchu obserwatora i każdemu z nich przypisujemy ilość klatek animacji, które mają się znajdować pomiędzy tym, a następnym punktem.

Ponadto dla każdego z tych punktów możemy ustawić następujące parametry:
"**East**" - położenie wschód-zachód, "**North**" - położenie północ-południe,
"**Alt**" - wysokość bezwzględna,
"**Dir**" - kąt skierowania kamery,
"**Pitch**" - odchylenie osi kamery od poziomu,
"**Bank**" - pochylanie kamery na boki,
"**Lens**" - ogniskowa obiektywu.

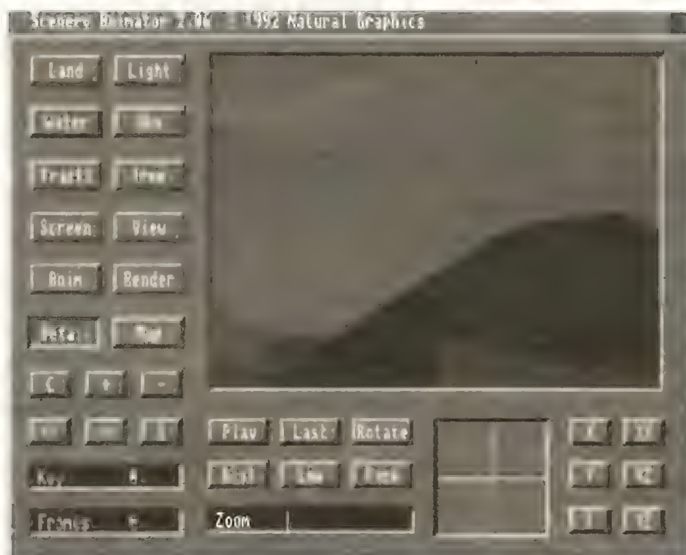
Ogniskową obiektywu ustawiamy na 18, co odpowiada dość dużemu kątowi widzenia, natomiast wysokość położenia obserwatora na 1700.

Parametry "Pitch" i "Bank" pozostawiamy wyzerowane (ciekawa jest funkcja automatycznego pochylania kamery "na zakrętach" ("**Auto banking**"), dostępna w menu "Options"). Póki przystąpimy do definiowania toru włączamy "**Curve**", co spowoduje, że nasz tor będzie opisany łagodną krzywą, a nie odcinkami prostymi.

Korzystamy także z funkcji "**DTan**", która automatycznie obraca kamerę w kierunku ruchu (dzięki temu nie musimy ustawiać parametru "Dir").

Jeśli nie dysponujemy dyskiem twardym, to w polu "**Frames**" wpisujemy 4, co oznacza, że każdemu punktowi "kluczowemu" przypisujemy 4 klatki animacji.

Posiadaczom dysków twardych polecam zwiększenie ich liczby, co da bardziej płynną, lub dłuższą animację. Punkty "kluczowe" możemy ustawiać na mapie za pomocą lewego przycisku myszy (kierunek i kąt patrzenia także możemy regulować bezpośrednio na mapie korzystając z prawego



przycisku), lub też wpisując ich współrzędne w polach "East" i "North".

Proponuję następujące ich ustawienie (Key - East, North):

1 - 3200, 6000,
2 - 3050, 5000,
3 - 3200, 4000,
4 - 4000, 4000,
5 - 4150, 5000,
6 - 4000, 6000.

Po wprowadzeniu współrzędnych każdego punktu klikamy na "**S**", aby zapisać to ustawienie, a następnie na "**+**" co zwiększa ich ilość.

Pomiędzy tymi punktami przemieszczać się możemy korzystając z gadget'ów "**<<**" i "**>>**" (lub wpisując numer w polu "Key"), ich liczbę zmniejszać możemy za pomocą "**-**", a skasować wszystkie wybierając "**C**".

Wszystkie zmiany dokonane później w danym punkcie zatwierdzamy za każdym razem korzystając z "S". Położenie punktów można dowolnie zmieniać, należy jednak pamiętać aby w polu widzenia kamery nie pojawiła się krawędź mapy, oraz aby nie zderzyć się z ziemią, o czym zresztą program nas ostrzega.

Po zakończeniu wprowadzania punktów klikamy na "**Lp**", co spowoduje zamknięcie naszej krzywej. Gdyby ktoś chciał obejrzeć rozmieszczenie poszczególnych klatek animacji wystarczy użyć funkcji "**Xpand**", która powoduje zamianę wszystkich klatek na "kluczowe". Warto skorzystać też z funkcji "**Pro**", która wyświetla widok toru z profilu - można łatwo ocenić, czy przebiega on na odpowiedniej wysokości od ziemi.

Jest też coś dla miłośników górskich jezior - funkcja "**Lake**" - wystarczy tylko wyszukać odpowiednią dolinę i kliknąć na jej brzegu. Gdy tor mamy już gotowy wracamy do głównego modułu programu klikając na "**Main**". Przed przystąpieniem do wyliczania końcowej animacji warto skorzystać z "**Play**" i obejrzeć jej wektorową postać. Można także wyliczyć dowolną klatkę ustawiając jej numer w polu "Key" i korzystając z funkcji "**Render**" (aby przyspieszyć tę operację wystarczy wyłączyć opcję "Detail").

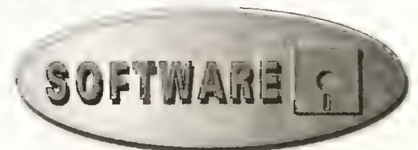
Następnie w module "Screen" należy wyłączyć interlace i wybrać rozdzielczość 320*256 (chyba, że mamy dysk twardy i dużo pamięci). Po powrocie do głównego modułu, w menu "**Anim Mode**" ustawiamy "IFF ANIM5", czyli standardowy format animacji i naciskamy "Anim" (włączając uprzednio "Detail" oczywiście).

Teraz zależnie od szybkości naszego komputera musimy trochę poczekać na efekty naszej pracy - na A500 potrwa to parę godzin. Standardowo Scenery Animator tworzy animacje w 32 kolorach - jest to dość mało, co niestety jest widoczne.

Oczywiście można to ograniczenie ominąć korzystając z możliwości zapisu przez program grafiki 24 bitowej. Gdy w menu "Anim Mode" format zapisu ustawimy na "IFF24 frames", to na dysku otrzymamy szereg plików, z których każdy zawiera jedną klatkę animacji. Wystarczy teraz dokonać ich konwersji na HAM za pomocą ADPro lub Deluxe Paint'a 4.5 i połączyć w animację formatu ANIM5. Posiadacze Amig wyposażonych w układy AGA zamiast trybu HAM użyć mogą HAM8 - naprawdę jest na co popatrzeć!

Konrad Dubiel

AUTOSTARTY na C-64



Prawdopodobnie większość z was w czasie pracy z C64 zetknęła się już z programami, które po wczytaniu samoczynnie się uruchamiały. Być może też niektórzy z was zastanawiali się w jaki sposób osiągnąć to we własnych programach

W tym artykule postaram się wyjaśnić w jaki sposób działają takie programy. Do pracy będzie potrzebny jakikolwiek monitor kodu maszynowego i odrobina znajomości assemblera procesora 6502.

Ze względu na to, że większość (niestety) posiadaczy C64 w Polsce używa go wraz z magnetofonem, zajmiemy się bliżej autostartami kasetowymi. Zawarte tu informacje, będą w 90% prawdziwe także i dla stacji dysków.

Żeby zrozumieć zasadę działania takich programów, musimy zapoznać się przynajmniej pobieżnie z systemem zapisu programów na kasecie.

Program kasetowy składa się z dwóch części: nagłówka oraz właściwego bloku danych. Obie są poprzedzone kilkusekundowym sygnałem synchronizującym. Rzeczą, która interesuje nas szczególnie jest nagłówek. Zawiera on m.in. tytuł programu, adresy wczytywania oraz bajt kontrolny, który określa typ wczytywanego pliku. Może on mieć następujące wartości:

- 1 - nagłówek pliku relokowalnego

- 3 - nagłówek pliku nierelokowalnego

- 4 - nagłówek pliku danych

- 5 - logiczny koniec zbioru

Szczególnie dla nas ważne są dwie pierwsze wartości. Cóż to bowiem oznacza, że plik jest relokowalny lub nie? Otóż plik reloko-

walny jest wczytywany zawsze pod aktualny adres startowy programów napisanych w BASIC'u zawarty w komórkach $2b,$2c, natomiast plik nierelokowalny jest zawsze ładowany pod adres podany w nagłówku.

Z tej właściwości KERNAL'a (systemu operacyjnego C64) korzystają właśnie programy samouruchamiające się. Ogólna zasada działania tych programów jest następująca: krótki, specjalnie przygotowany nierelokowalny plik wczytywany jest w obszar wektorów systemowych.

W programie umieszczone są nowe wartości niektórych wektorów, ustawione na adres początkowy programu ładującego dalszą część programu (program ten jest zwykle umieszczony zaraz za zmiennymi systemowymi). Zazwyczaj modyfikowane są następujące wektory:

1.$0302,$0303 - adres pętli interpretera
2.$0314,$0315 - wektor przerwań irq
3.$029f,$02a0 - miejsce w którym system przechowuje zawartość wektora irq w czasie ładowania (system wykorzystuje wtedy ten wektor do własnych celów).

Do programów startujących za pomocą wektora nr 1 możemy zajrzeć stosunkowo łatwo. Wystarczy wgrać je z poziomu dowolnego debuggera i dopóki nie wrócimy do BASIC'a, to program się nie uruchomi i możemy go spokojnie badać.

Gorzej gdy program startuje za pomocą wektora nr 2 lub nr 3. Tu samo wczytanie z poziomu monitora nie skutkuje i program w dalszym ciągu się uruchamia. Wynika z tego, że autostart za pomocą tych wektorów jest znacznie lepszy, ale oczywiście na wszystko jest rada. Poradzić sobie możemy na dwa sposoby:

1.poprawić oryginalny nagłówek programu tak, aby otrzymać plik relokowalny.
2.napisać specjalny program wczytujący pliki pod dowolny adres.

Ja wybrałem to drugie rozwiązanie. Program okazał się bardzo krótki, a co najważniejsze w 100% skuteczny. Skorzystałem w nim z kilku procedur zawartych w ROM'ie lecz nie zostały one wywołane od początku, mój program skacze "do środka" tych procedur, aby ominąć pewne zbędne fragmenty kodu.

Program wczytuje każdy plik (relokowalny lub nie) pod adres $0801. Można go zmienić ustawiając odpowiednie wartości pod adresami $c00f i $c011. Po wczytaniu można program dowolnie modyfikować, a następnie przepisać go w odpowiednie miejsce i uruchomić. W komórkach $c0f0 - $c0f3 zapamiętywany jest właściwy adres początkowy i końcowy ładowanego pliku. W przypadku wystąpienia błędu podczas wczytywania program powróci do monitora (lub BASIC'a w zależności od sposobu wywołania) ustawiając kolor ramki na czarny.

No a jak tworzyć programy z autostartem??? To już zadanie dla was, posiadacie wystarczającą ilość informacji by tego dokonać. Jeżeli wam się to uda, to prześlijcie swoje programy do naszej redakcji. Najlepsze z nich opublikujemy!

Krzysztof "BRUSH" Dąbrowski

O wektorach jeszcze raz

SOFTWARE

Wypełniane figury pojawiły się już dosyć dawno w komputerowym światku i na dobre zadomowiły się w demach. Kiedyś były sztandarowym produktem grup, które bardzo chciały liczyć się na scenie

Wbrew pozorom, wektorówka pojawia się w każdym liczącym się demku, tyle tylko, że jest coraz bardziej wyrafinowana, można powiedzieć: dla smakoszy. A to coraz więcej ścian, a to jednocześnie dwie figury rzucające cień na siebie, oświetlane przez poruszającą się lampkę, czasem nudzący się koderzy dodają na ścianach sześcianu kilkukolorowe obrazki lub kolejne obracające się figury...

Jak się w tym wszystkim połapać? Wektorówkę można podzielić na dwa podstawowe typy: dokładną i niedokładną.

Pierwszy typ ma tą zaletę, że możemy zdefiniować praktycznie każdy rodzaj figury (sześcian, statek kosmiczny, twarz mamy, ludzką postać, etc.), trzeba się jednak liczyć z nakładami czasowymi, bowiem im bardziej skomplikowany kształt, tym wolniejsza będzie jego animacja. Animacja ma zaś to do siebie, że większa częstotliwość ukazywania się kolejnych faz ruchu przyczynia się raczej do zwiększenia niż zmniejszenia jej atrakcyjności. Właśnie po to wymyślono drugi rodzaj wektorówki, szybkiej, aczkolwiek z dosyć prostymi figurami. Teraz chciałbym nieco przybliżyć ten drugi typ, jako bardziej przystępny i zrozumiały.

Na początek o wymaganiach

i umiejętnościach. Wręcz niezbędna jest znajomość komend procesora Motorola 68000, mniej, aczkolwiek przyda się podstawowa choć wiedza o blitterze i copperze (nasze wektory trzeba jakoś oglądać). Dobrze by było mieć jakikolwiek assembler, bo kodowanie na kartce z kalkulatorem w ręku, a po jakimś czasie i w głowie, może wykończyć nawet największych zwolenników komputerów.

Punkt pierwszy:
z czym to się je?

Zacznijmy od sześcianu. Jak sama nazwa wskazuje, ma toto 6 ścian i nic poza tym. Jeśli nie spaliście na matmie, to wiecie, że wierzchołki bryły przestrzennej przedstawia się za pomocą 3 współrzędnych (x,y,z).

By zdefiniować sześcian potrzebnych będzie 8 wierzchołków. Pierwsze cztery tworzące kwadrat na płaszczyźnie XY:

```
(-100,100,-100),
(100,100,-100),
(100,-100,-100),
(-100,-100,-100).
```

Pozostałe mają te same współrzędne x i y, ale ich współrzędną z ustawiamy na 100. Ale same pojedyncze punkty niczego nam jeszcze nie dają. Ponieważ będziemy rysować ściany, musimy wiedzieć, która ściana składa się z jakich punktów.

Do naszego pliku źródłowego dołączamy więc odpowiednią tablicę, zbudowaną na przykład tak:

a - numer koloru ściany,
b - ilość wierzchołków ściany,
c, d, e, f - spis wierzchołków w kolejności rysowania linii.

W naszym przykładzie owa tablica może wyglądać następująco:

```
1, 4, 1,2,3,4
1, 4, 8,7,6,5
2, 4, 2,6,7,3
2, 4, 5,1,4,8
3, 4, 2,1,5,6
3, 4, 3,7,8,4
```

Jeśli przyjrzymy się uważnie tej tablicy to dojdziemy do paru interesujących wniosków. Po pierwsze, używamy tylko 3 rejestrów kolorów mimo, że określiliśmy 6 ścian. Można tak postąpić, gdyż będziemy rysować tylko widoczne ściany, a takich w naszej figurze są maksymalnie 3. Po drugie, ten sam rejestr koloru przypisaliśmy ścianom przeciwległym - niewidocznym jednocześnie. Trik ten stosuje się przy bryłach foremnych lub o parzystej liczbie ścian. Jest jeszcze jedna ważna sprawa.

Ponieważ mamy rysować tylko widoczne ściany, trzeba było je jakoś odróżnić od siebie. Weźmy jako przykład dowolny wielokąt. Gdy potraktujemy jego krawędzie jako wektory, to jesteśmy w stanie narysować go na płaszczyźnie w jednym, wybranym przez nas kierunku.

Warto w tym momencie dokonać eksperymentu na jakimś padniętym dysku. Łapiemy go mocno i obracamy wokół osi y (inaczej mówiąc, przy obrocie, dziurka protekcji dysku porusza nam się w kierunku poziomym, jeśli jest inaczej, wskazane byłoby udać się do okulisty).

Zauważmy, że po obróceniu o kąt większy niż 90 stopni, wierzchołki dysku, które poprzednio ustawione były zgodnie z ruchem wskazówek zegara, teraz ustawione są przeciwnie. Wyobraźmy sobie, że obracamy sześcian, który składa się właśnie z takich dysków. Gdy zdaży się taka sytuacja, że ściana jest gdzieś z tyłu (jest niewidoczna), to możemy być pewni zmiany jej kierunku rysowania.

Punkt drugi:
jak obracać ?

Odpowiednie wzory na obroty w trzech wymiarach i przekształcenie do dwóch wymiarów zostały podane w Kebabie 6/92.

W językach wyższego poziomu (C, Pascal, Modula, Basic,...) zastosowanie tych wzorów to sprawa tylko dokładnego ich przepisania, w Asemblerze na Amidze można oczywiście stosować procedury zawarte w ogólnie znanych bibliotekach matematycznych, jest to jednak zupełnie nieopłacalne czasowo. Okazuje się, że funkcje sinus i cosinus można spokojnie obliczyć nie korzystając z bibliotek.

Tu przydają się wszelkiego rodzaju sinusmaker'y, czyli programy tworzące tablice sinusów z kolejnymi wartościami tej funkcji.

Podajemy takie parametry początkowe: okres - 1024, amplituda - 65535, wartość początkowa - 0.

Zostanie wygenerowana tablica od -32767 do 32766. Teraz aby pomnożyć jakąś liczbę przez sinus jakiegoś kąta, wykonujemy takie oto proste rozkazy:

```
muls d7,d0     ; d0 - np. wsp. x,
               ; d7 - wartość po-
               ; brana z tablicy

add.l d0,d0

swap  d0       ; d0 = x*sin(a)
```

Dwa ostatnie rozkazy odpowiadają podzieleniu rejestru d0 przez 32768. W tym momencie trzeba wytłumaczyć te dosyć duże liczby.

Ponieważ funkcja sinus ma wartości w zakresie -1 do 1, a procesor M68000 nie operuje na liczbach zmiennoprzecinkowych, trzeba jakoś z tej niedogodności wybrnąć (w tym przypadku lepiej nie kupować koprocesora matematycznego...).

Po prostu mnożymy i dzielimy przez tą samą liczbę, w naszym przypadku 32768. Ze szkoły podstawowej wiemy, że wtedy nie zmienia nam się wartość wyrażenia. Dla nas ważna jest otrzymana dokładność, co wydatnie wpłynie na płynność obrotów figur.

Podobnie można postąpić przy wzorze na perspektywę. Tym razem do tablicy zapisujemy wartości wyrażenia [($7fff*Zv) / Z]

dla kolejnych wartości Z większych od Zv, czyli współrzędnej z obserwatora.

Oczywiście należy wybrać jakiś rozsądny rozmiar tej tablicy. Z reguły jest to kilkanaście kilobajtów.

Przy obliczaniu perspektywy wykonujemy rozkazy:

```
and.w #-2,d2

muls  (a0,d2.w),d0  ; a0- adres startowy
                    ; naszej tablicy

swap   d0           ; d0,d1,d2- współrzęd-
                    ; ne x,y,z punktu

muls    (a0,d2.w),d1

swap   d1
```

Operacja and jest niezbędna, gdyż w tablicy mamy zapisane dane w postaci słów i wobec tego możemy operować tylko na parzystych adresach. Ponieważ jak już wspomniałem będziemy rysować tylko widoczne ściany, musimy jakoś rozstrzygnąć które rysować. Obliczamy mianowicie podane niżej wyrażenie i badamy jego znak. Jeżeli wynik jest ujemny, ściana jest obrócona do nas tyłem i jest niewidoczna (inaczej mówiąc zmienił się jej układ z prawoskrętnego na lewoskrętny).

```
W=(X1-X2)*(Y3-Y2)-(X2-X3)*(Y2-Y1)
```

Jako (X1,Y1), (X2,Y2), (X3,Y3) przyjmujemy 3 kolejne wierzchołki z badanej ściany obliczone ze wzoru na perspektywę.

Punkt trzeci:
wreszcie konkrety.

W wyniku ostatniego przekształcenia otrzymaliśmy normalne współrzędne X i Y uwzględniające odległość obiektu od obserwatora. Co zrobić by nasz sześcian wreszcie zaczął się ruszać? Schemat postępowania jest prosty:

1.Uruchamiamy copper tak, aby na ekranie oglądać wybrany wycinek pamięci w którym to wycinku rysować będziemy sześcian;
2.Czyścimy ekran po poprzedniej fazie animacji;

3.Obliczamy współrzędne figury;

4.Blitterem rysujemy tylko widoczne ściany;

5.Wypełniamy obszar w którym narysowaliśmy linie;

6.Powrót do punktu 2.

Ze względu na sposób wypełniania obszaru przez blitter trzeba zastosować pewien trik. Otóż linie rysujemy zawsze w jednym kierunku (to jest w górę lub w dół), a przed obliczeniem parametrów linii dla blittera odejmujemy 1 od współrzędnej y punktu końcowego linii. Zapobiega to skutecznie dwóm dosyć nieprzyjemnym efektom: rozchodzeniu się ścian i wystąpieniu nieparzystej ilości punktów w linii.

Nie można zapomnieć także o podaniu parametrów dla blittera specyficznych dla pracy w trybie wypełniania. Przy rysowaniu linii podajemy minterm (w rejestrze bltcon0) równy $4a zamiast $ca oraz ustawiamy bit SING w bltcon1.

To by było na tyle jeżeli chodzi o obiekty niewypukłe. Cóż, w normalnym, ludzkim świecie nie jest ich dużo. Dlatego też w wielu demach pojawił się nowy sposób rysowania figur w wersji dużo bardziej dokładnej i umożliwiającej obrazowanie ciekawszych obiektów (notabene jest on używany we wszystkich symulatorach lotu). Jeżeli potrafimy już wyświelić i poobracać proste figury to nic nie stoi na przeszkodzie aby nasz programik nieco zmodyfikować.

Zastanówmy się najpierw dlaczego poprzednie procedury mogły stworzyć poprawny obraz sześcianu, a nie potrafiły uporać się dajmy na to z wahadłowcem Columbia. Ano dlatego, że dwie ściany narysowane na tym samym bitplanie i nałożone na siebie kolidowały ze sobą i nie były zachowane poprawnie kolory. Aby temu zapobiec, musimy używać bufora, czyli wycinka pamięci, w którym będziemy rysować ściany, a dopiero potem przenosić je na ekran. Rozważmy taką przykładową płaszczyznę, którą chcemy narysować tak by jej kolor był określany przez rejestr COLOR6. Liczba 6 wskazuje nam, że ściana ma być narysowana na bitlane'ach 2 i 3, a bitplane 1 jest pusty. Wynika to oczywiście z binarnego przedstawienia liczby 6 = %110. Dla nas bit0 to bitplane 1, bit1 to bitplane 2 itp.

Przy kodowaniu musimy użyć operacji OR z naszym pomocniczym buforem i bitplane'ami 2 i 3, a bitplane nieużywany (w naszym przypadku pierwszy) czyścimy operacją AND z buforem jako mas-



ką. To jednak nie wystarczy. Wyobraźmy sobie sytuację, gdzie nakładają nam się 2 ściany i czasem jedna jeśt bliżej, a czasem druga. I co teraz ?

Musimy niestety posortować ściany w zależności od ich odległości od ekranu i jako pierwsze rysować te położone najdalej. Najefektywniej można je posortować ze względu na współrzędne zetowe wszystkich wierzchołków tworzących ścianę, które uśredniamy (dodajemy wszystkie do siebie i dzielimy przez ilość). Gdy wykonamy już te wszystkie operacje, pomocniczy bufor należy przed jego kolejnym wykorzystaniem wyczyścić, najlepiej za pomocą blittera.

W ten dość prosty sposób możemy już cieszyć oczy statkami kosmicznymi, samochodami itp. Żeby jednak stworzyć coś okazalszego, np. wnętrze jakiegoś pomieszczenia czy tzw. vector world, należy się jeszcze sporo pomęczyć. Przede wszystkim musimy opanować obcinanie odcinków tak aby nie rysować wszystkiego co istnieje w naszym nibyświecie, a jedynie to co aktualnie widzi obserwator.

Obcinanie (ang. clipping) nie jest nie do zrealizowania, trzeba tylko pamiętać, że jeżeli odcinek jest "ucięty" z prawej strony, należy dorysować pionową linię w miejscu ucięcia. Bez tego zabiegu, wektorówka przeistoczy nam się w dziwne, awangardowe efekty, ewentualnie obejrzymy swojsko mrugającą, czerwoną ramkę. Warto pokusić się także o realizację oświetlenia. Najprostsza metoda to obliczanie poziomu jasności ściany w zależności tylko od współrzędnych zet wierzchołków ją tworzących. Znajdujemy najmniejszy i największy zet spośród nich, odejmujemy od siebie i dzielimy przez odpowiednio dobrany parametr. Jeśli wynik jest mniejszy od zera, to kolor ściany jest zero, natomiast jeśli jest większy od 15 to przyjmujemy największą jasność tegoż koloru.

Możemy oczywiście pobawić się z oświetleniem ruchomym, należy wtedy badać kąt między wektorem normalnym do powierzchni ściany a wektorem poprowadzonym od źródła światła do początku wektora normalnego. Bardzo efektowne jest także wyznaczanie cieni, łatwo to zrobić gdy cienie pojawiają się tylko na podłożu, bo znacznie trudniej jest znaleźć ich obraz na wszystkich, aktualnie widocznych ścianach.

Liczba możliwych efektów związanych z szeroko rozumianymi wektorami jest duża i właściwie każde nowe demo wnosi coś nowego do tego działu komputerowej animacji. Niektórym może się wydawać, że wektory służą tylko do nudzenia i tak już znudzonego, przeciętnego oglądacza dem. Od tego jednak rozpoczęło się tworzenie sztuki przy pomocy maszyny, a od epoki wypełnianych figur homo sapiens przeszedł żmudną drogę na końcu której znajduje się system Silicon Graphics no i oczywiście nasze kochane raytracery: IMAGINE i REAL 3D.

Robin/WFMH

ZE SCENY
ELYSIUM

Grupa powstała w lutym 1992 (w czasie ferii zimowych). Założycielami byli **BRUSH**, **HAIN** i **SKY** (obecnie **BIONDI**), którzy wspólnie zdecydowali się opuścić PARADOS (na skutek różnych problemów wewnętrznych) i założyć swoją grupę. Jako, że SKY przygotowywał się właśnie do matury, więc był chwilowo nieaktywny i cały trud zakładania grupy spadł na barki HAIN'a i BRUSH'a.

Udali się oni w podróż po Polsce w czasie której odwiedzili najlepszych (ich zdaniem) ludzi ze sceny i dwóm z nich (**CARRION**'owi i **U-PUKE**'owi/ASPHYXIA) zaproponowali wstąpienie. Pierwszy zgodził się po kilku dniach namysłu, drugi odszedł ze sceny, więc propozycja przestała go dotyczyć. Do grupy dołączyli też dwaj muzycy **LONGHAIR** i **TOULDIE**. Zaraz po tym oficjalnie ogłoszono powstanie grupy i wiądomość tą rozpowszechniono za pomocą wszystkich scenowych magazynów. Wszyscy rozpoczęli mniej lub bardziej intensywne prace nad demem i grupa powoli nabierała rozmachu.

Ukazały się też w tamtym okresie dwa numery legendarnego już dziś magazynu *HIGHLIFE* (nr.11 i 12). Na copy-party w Warszawie dołączył nowy grafik **GRYF** i przez dłuższy okres czasu grupa pracowała w tym właśnie składzie. Aktywność sięgnęła szczytu na wakacjach (jak to zwykle) i właśnie wtedy powstało demo *"ORIGON the request"*, które wygrało demo competition na ASPHYXIA party. Grupa cały czas się rozbudowywała i wzbogaciła się o dwóch członków z Niemiec (**ZORE** i **GASTON**) oraz o 4 członków z Polski (**JUMBO**, **DUXE**, **KUBA** i **CRUISE**). Obecnie ELYSIUM wydaje dwa magazyny: anglojęzyczny *INQUIRY* i polskojęzyczny *ALWAYS*. Obydwa debiutowały dopiero niedawno, ale wywołały spore poruszenie i wydaje mi się, że przypadły scenowiczom do gustu. Kolej teraz na kilka słów o członkach grupy:

BRUSH - samozwańczy leader, leniwy koder (chociaż z mnóstwem pomysłów). Właśnie skończył kodować wspólnie z LONGHAIR'em edytor muzyczny, który wkrótce bedzie w sprzedaży, pracuje teraz nad nowym demem, które prawdopodobnie ukaże się na LIGHT/HORIZON PARTY w Szwecji.

HAIN - bez wątpienia najlepszy cracker w Polsce. Jego hobby to

przerabiane wszelkiej maści packerów by dawały lepsze wyniki niż dają oraz ściganie sie z zaprzyjaźnionymi crackerami w konkurencji zwanej pod nazwą "kto-zrobi-lepszą-wersję". Czasami (jak się go dobrze przyciśnie) koduje jakieś efekty do dema.

BIONDI - leniwy grafik (zresztą jeden z pierwszych na polskiej scenie C64), były redaktor magazynu HIGHLIFE, aktualny redaktor magazynu ALWAYS, jego hobby (pospołu z CARRION'em) to wymyślanie "design'u" do dem. Najstarszy w grupie.

CARRION - aktualnie najlepszy grafik w Polsce (wystarczy spojrzeć na jego rysunki...), zapalony "design'owiec" (jego naczelne hasło to:"po co ci te scrolle??"), jego listy potrafią rozbawić nawet największego ponuraka (tylko ma ostatnio problem, bo połowa listów do niego niestety ginie), od ciągłego wlepiania oczu w ekran dorobił się okularów.

GRYF - posiadacz bez wątpienia najdłuższych włosów na scenie, grafik specjalizujący sie w animacjach i całoekranowych rysunkach, zapalony gracz (ulubione gry to np. "Psycho pigs"...), gdy na copy-party ludzie szaleją, piją, kończą dema, to on najspokojniej przegląda dyski kolegów w poszukiwaniu nowych (i starych) gier.

CRUISE - człowiek, który dosłownie "wyrósł spod ziemi". Nikt nic o nim nie słyszał (jego grupa nic nie robiła, więc nikt nie widział jego prac) i nagle pewnego pięknego dnia przysłał do BRUSH'a 3 obrazki we FLI. Z miejsca został przyjęty i teraz cały czas rysuje, rysuje i rysuje (to chyba najbardziej pracowity członek grupy).

ZORE - grafik i mega-swaper z Frankfurtu. Specjalizuje się w logach i fontach, obecnie trawi go nowa obsesja tj.redagowanie magazynu (INQUIRY). LONGHAIR - jeden z pierwszych (obok TRACKER'a) muzyków na Polskiej scenie. Zdegustowany poziomem dostępnych edytorow, sam sobie napisał player'a i teraz może wyczyniać co tylko mu się zamarzy. Obecnie (co słychać w jego muzyczkach) jest zafascynowany jazz-rockiem.

TOULDIE - wolny strzelec, większość czasu spędza przy swojej AMIDZE, ale jeżeli tylko potrzebna jest nam jakaś muzyczka, to siada do C64 i wkrótce powstaje coś co większości ludzi na scenie podoba sie najbardziej (o czym świadczy jego pozycja w charts'ach).

GASTON - muzyk z Drezna, o którym prawie nic nie wiadomo. Nigdy go nie widzieliśmy, wiemy o nim tylko trochę strzępów informacji ale jego utwory są świetne, a o to przecież chodzi.

JUMBO - koder z zacięciem do narzędziówek, obecnie zaangażowany w prace nad grą, bardzo niechętnie odnosi się do dem (oglądać lubi, kodować nie...), większość prac wykonuje na zamówienie lokalnego studia komputerowego. Jedyny w grupie spec od hardware'u.

DUXE - megaswaper, którego kłopoty z pocztą stają się powoli legendarne, redaktor magazynu ALWAYS, karateka.

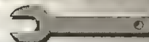
KUBA - swaper, czyli nasze "okno na świat", szybko podąża do upragnionej setki kontaktów.

MASTER JAY - najnowszy nabytek w grupie, megaswaper (150 kontaktów) z Niemiec, próbuje też swoich sił w grafice.

Krzysztof "BRUSH" Dąbrowski

Budowa modemu do pracy Packet Radio

HARDWARE



Dziś gratka dla wszystkich zainteresowanych podłączeniem swoich radiostacji do swoich komputerów na potrzeby pracy z Packet Radio. Osoby posiadające wystarczające zasoby finansowe mogą spokojnie kupić specjalizowane modemy np. TNC-2. Mniej zasobnym polecamy przeczytanie tego artykułu, po lekturze którego oraz po pomocy doświadczonego kolegi-elektronika, stać się można posiadaczem sprawnego modemu do Packet Radio.

Obecnie modem do tych zastosowań można zrealizować na trzy sposoby (najbardziej popularne):

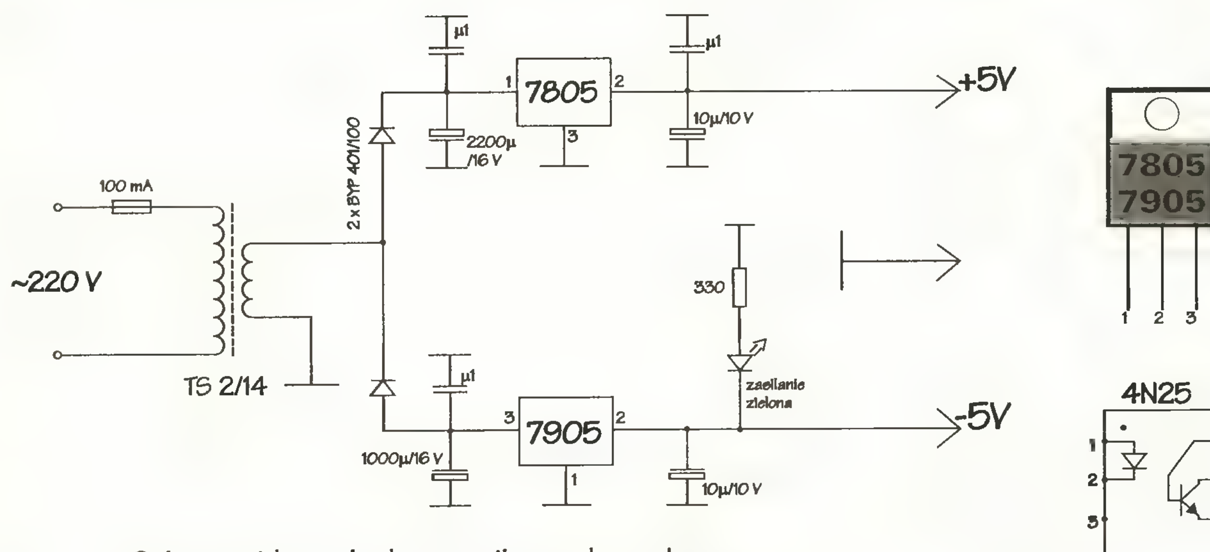
1. Wykonując go na układach XR 2211 i XR 2206. Jest to najtańsza wersja układu, pracująca poprawnie, ale okazuje się, że można ją jeszcze poprawić np. przez zastosowanie selektywnych filtrów pasmowo-przepustowych, co znacznie komplikuje budowę tego układu.

2. Modem można wykonać także tylko na jednym układzie scalonym - TCM 3105. Konstrukcja ta jest bardzo oszczędna pod względem ilości zastosowanych elementów i ilości pobieranego prądu, ale jest stosunkowo droga a w/w układ scalony jest bardzo trudno dostępny.

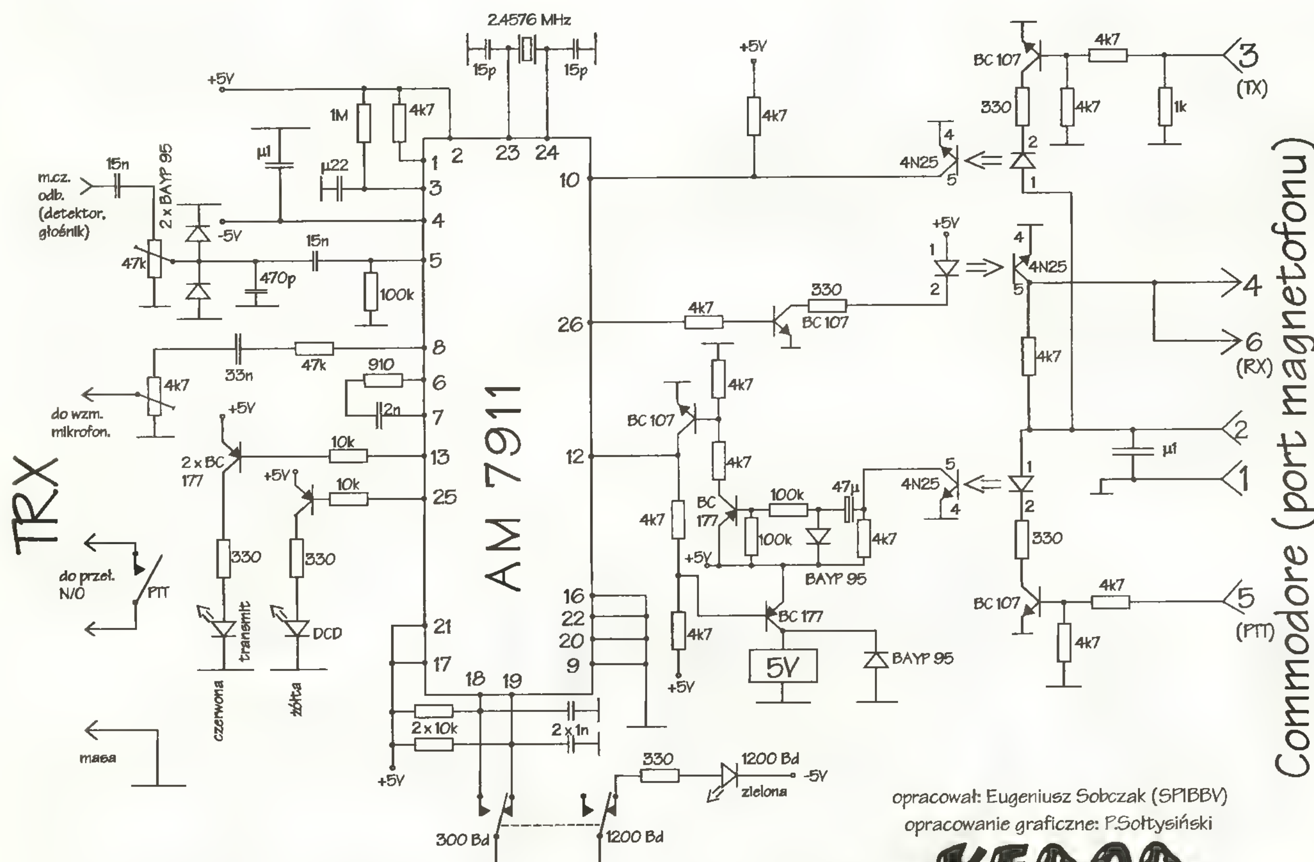
3. Poniżej opisane rozwiązanie jest również mało skomplikowane pod względem ilości zastosowanych elementów a także (co jest istotne) zapewnia wysoki poziom bezpieczeństwa dla pracy komputera.

Opisany układ nadaje się do pracy z komputerami klasy PC (z programami emulującymi TNC) np.

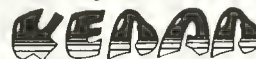
Schematy ideowe zasilacza i modemu Packet-Radio dla programu DigiCom 64



Schemat ideowy budowy zasilacza do modemu.



Schemat ideowy modemu Packet Radio dla Commodore 64



Baycom; do komputera Atari oraz dla komputerow rodziny Commodore (C-16, C+4 i C-64). Cała konstrukcja jest oparta na układzie scalonym AM 7911. Jest to specjalizowany układ produkowany do konstruowania modemów (także telefonicznych - opcjonalna praca w duplexie).

W komplecie z powyższym układem należy zakupić także rezonator kwarcowy o częstotliwości 2.4576 MHz, który pracuje jako zegar taktujący. Opisane wykonanie modemu jest najprostsze, ale jednocześnie wykonane tak, aby zapewnić jak największe bezpieczeństwo dla pracy naszego komputera oraz dużą uniwersalność pod względem przyłączania urządzeń nadawczo-odbiorczych.

Pewną niedogodnością pracy z układem AM 7911 jest konieczność doprowadzenia dwóch napięć zasilających: +5 V i -5 V. Komplikuje to trochę jego zastosowanie w sprzęcie przenośnym przy zasilaniu np. z instalacji elektrycznej samochodu. Modem może pracować przy szybkości transmisji 1200 Bd na UKF i 300 Bd na KF. Do cech pozytywnych omawianego układu należy zaliczyć między innymi fakt wyposażenia go w cyfrowe filtry m.cz. zapewniające poprawny odbiór PR (Packet Radio) nawet przy mocno zakłóconym sygnale lub przy bardzo słabym stosunku S/N (sygnał / szumy).

Sygnał z wyjścia m.cz. radiotelefonu o poziomie od 30 mV do 5 V podany jest przez potencjometr i diody zabezpieczające wejście układu scalonego na jego 5 nogę. Sygnał wejściowy o odpowiednich częstotliwościach (np. dla transmisji 1200 Bd są nimi 1200 Hz i 2200 Hz) zostaje odpowiednio zdekodowany i w efekcie otrzymujemy odpowiednie stany logiczne - 1 lub 0. Odbiór sygnału jest również sygnalizowany optycznie przez zapalenie żółtej diody LED. Z kolei, gdy dokonywana jest transmisja danych z naszego komputera to sygnały 0 i 1 przychodzące do 10 nogi układu AM 7911 powodują pojawienie się na jego wyjściu (noga nr 8) sygnału analogowego o częstotliwościach 1200 Hz i 2200 Hz, który jest podawany poprzez potencjometr na wejście mikrofonowe radiotelefonu.

Równocześnie komputer wysyła sygnał włączenia nadawania naszego radiotelefonu. W opisanym modemie zastosowany został układ włączania PTT (Push To Talk) z czasowym, samoczynnym odłączaniem nadawania (po około 35 sekundach). Czas ten został tak dobrany, że przy normalnej pracy programu PR jest w zupełności wystarczający ale w przypadku np. zawieszenia się programu nie będzie możliwe długotrwałe nadawanie, a tym samym - blokowanie kanału innym użytkownikom. Nadawanie sygnalizowane jest zapaleniem się czerwonej diody Led na obudowie modemu.

Jak widać na schemacie, na wyjściu z modemu do komputera zastosowana została izolacja galwaniczna połączenia, którą zrealizowano na transoptorze. Nasz modem jest bowiem podłączony z radiotelefonem, ten z kolei z umieszczoną na zewnątrz budynku anteną. Antena, szczególnie gdy nie jest uziemniona, narażona jest na na zbieranie ładunków elektrostatycznych, ładunków z wyładowań atmosferycznych, a to mogłoby spowodować awarię komputera. Dzięki zastosowaniu transoptora połączenie to jest rozdzielone galwanicznie.



Opisany układ modemu po zmontowaniu jest bardzo prosty do uruchomienia. Regulacje ograniczją się jedynie do wyregulowania poziomu sygnału analogowego m.cz. wychodzącego do wzmacniacza mikrofonowego radiotelefonu. Dopasowanie tego poziomu jest dosyć ważne dla poprawnego odbioru naszego sygnału PR przez innych respondentów. Musi on być tak nastawiony, żeby układ ogranicznika amplitudy we wzmacniaczu mikrofonowym jeszcze go nie zniekształcił (obciął) ale jednocześnie modulacja nadajnika nie stała się zbyt słaba. Ustawienie poziomu sygnałów z odbiornika do modemu nie jest krytyczne, może on znajdować się w szerokim zakresie napięć: od 30 mV do 5 V.

Do zasilania modemu trzeba wykonać zasilacz dostarczający dwóch napięć niezbędnych do pracy układu AM 7911. Zasilacz został wykonany na transformatorze sieciowym TS 2/14 i dwóch stabilizatorach zintegrowanych: 7805 dla napięcia +5 V i 7905 dla -5 V.

Modem posiada przełącznik szybkości pracy: 1200 Bd i 300 Bd. Przy pracy na KF z użyciem modulacji jednowstęgowej SSB trzeba by było jeszcze dorobić do modemu wskaźnik precyzyjnego dostrojenia do odbieranego sygnału.

Opisywany modem możemy przyłączyć do Commodore 64 przez port magnetofonu. Na schemacie zaznaczono kolejność wyprowadzeń tego gniazda. Do pracy z tym modemem podłączonym do Commodore 64 musimy zaopatrzyć się w odpowiedni program, np. DigiCom 64. Wymaga on

współpracy ze stacją dysków. Tym, którzy jej nie posiadają, do dyspozycji pozostawiono bardzo niewielkie możliwości i liczenie na "litość" kontaktujących się z naszym komputerem osób, bo a nuż któraś z nich zechce np. obejrzeć katalog naszego dysku? Strach pomyśleć...

Eugeniusz Sobczak - SP1BBV

Zapomniane Urządzenia - cz. 2



Zgodnie z zapowiedzią przejdziemy teraz do właściwego tematu tego artykułu. Pierwszym z urządzeń, które chciałbym tu omówić jest NIL:

Urządzenie to różni się od pozostałych, gdyż... w zasadzie nic nie robi. Piszę "w zasadzie", gdyż jego jedynym zadaniem jest pochłanianie danych do niego skopiowanych. W praktyce NIL: wykorzystuje się do likwidowania komunikatów wypisywanych przez programy do okienka CLI. Załóżmy, że po każdym zabootowaniu dyskietki, chcemy uruchomić program RexxMast. Jeżeli w sekwencji startowej (startup-sequence) wpiszemy linię:

```
RexxMast
```

to owszem, ów program uruchomi się, ale jednocześnie wypisze na ekranie komunikaty o autorze, wersji, itp. Gdybyśmy te napisy chcieli zlikwidować to zamiast powyższego RexxMast, wpiszmy:

```
RexxMast >NIL:
```

Z poprzedniego odcinka wiemy, że linia ta uruchomi nasz program z jednoczesnym przełączeniem standardowego wyjścia na urządzenie NIL: Dlatego wszystkie komunikaty nie będą pokazywały się w oknie, tylko zostaną skierowane do NIL: i w związku ze specyfiką tego urządzenia zginą bezpowrotnie. Takie rozwiązanie wiąże się z pewnymi niedoskonałościami, ponieważ, gdy podczas uruchamiania naszego programu zaistnieją błędy, to komunikaty ostrzegające bądź informujące o zaszłym niepowodzeniu, również zostaną skopiowane do NIL:, więc w konsekwencji użytkownik nie będzie mógł ich odczytać.

Przejdźmy może do bardziej interesującego urządzenia, mianowicie SPEAK:

Aby móc z niego korzystać należy najpierw uczynić go widzialnym dla systemu. Dokonuje się tego komendą Mount. Wpisujemy zatem:

```
Mount SPEAK:
```

i po kłopocie - już możemy z niego korzystać. Proszę teraz przekręcić pokrętło regulacji głośności na nieco wyższy poziom niż zwykle i wpisać komendę:

```
copy  S:startup-sequence
SPEAK:
```

która to kopiuje plik startup-sequence do naszego urządzenia. I jaki efekt? Komputer czyta zawartość owego pliku! Jeśli komuś to nie wystarcza proszę spróbować w ten sposób:

```
dir >SPEAK: df0:
```

i Amiga powinna powiedzieć co znajduje się w katalogu głównym dyskietki aktualnie włożonej do stacji. Jak już łatwo się domyślić skopiowanie jakiegokolwiek pliku tekstowego do urządzenia SPEAK: spowoduje, że komputer przeczyta i wypowie jego zawartość.

Następnym urządzeniem na które warto zwrócić uwagę jest PIPE: Zainstalujmy je zatem komendą:

```
Mount PIPE:
```

Urządzenie to służy do przekazywania dużych zbiorów danych pomiędzy programami. Najlepszym przykładem praktycznego jego zastosowania jest współpraca i wymiana informacji na linii PowerPacker - CygnusEditor.

Wszystkie dotychczas proponowane (również w Kebabie) rozwiązania ładowania skompresowanych tekstów polegały na tym, że PowerPacker dekopresował tekst, zapisywał go na ramdysku, a stamtąd dopiero odczytywał go CygnusEditor. Rozwiązanie to nie jest zbyt opłacalne ze względu na jego wymagania pamięciowe, zauważalne w przypadku dłuższych tekstów.

I tutaj właśnie przychodzi nam z pomocą PIPE: Otóż jeśli zdekompresowany tekst zapiszemy PowerPackerem pod nazwą np. `PIPE:TempText` to przesłana zostanie tylko jego część (w przypadku długiego tekstu) i PowerPacker zatrzyma się, i będzie stał dopóki jakiś inny program nie odczyta pliku TempText z urządzenia PIPE: Jak widać urządzenie to stanowi jakby kanał pośredniczący w wymianie danych. Zaletą tego rozwiązania jest brak konieczności tworzenia plików pośredniczących, a co za tym idzie oszczędność pamięci.

Kolejnym urządzeniem jest RAD: Posiada ono podobne zadanie do zwykłego RAM:, mianowicie służy do czasowego przechowywania danych. Różnica między nimi polega głównie na tym, że zawartość RAD: nie jest tracona po zresetowaniu komputera. Poza tym zwykły RAM: zajmuje tyle pamięci ile pliki w nim się znajdujące, natomiast pamięć dla RAD: rezerwowana jest w momencie jego instalowania komendą:

```
Mount RAD:
```

RAD: swoją budową niezmiernie przypomina zwykłą dyskietkę. Tak jak DF0: charakteryzuje się podziałem na sektory, można z niego zabootować komputer, oraz komendą DiskCopy skopiować "po trakach" całą zawartość dyskietki.

W związku z powyższym zupełnie swobodną pracę z RAD: można uzyskać dopiero na systemach zaopatrzonych w minimum 2MB pamięci. Pracę z RAD:'em kończymy wydaniem komendy:

```
RemRad
```

UWAGA! Obecność RAD:'dysku mogą sygnalizować niektóre Virus-Protectory komunikatem, że w pamięci jest wirus. Oprócz tego w wyniku błędu w systemie operacyjnym 1.3 RAD:'dysk jest niszczony po resecie w przypadku posiadania więcej niż 512KB pamięci CHIP. Komenda:

```
SetPatch r
```

podana najlepiej w Startup-Sequence usuwa w/w błąd.

Krzysztof Kobus

Asembler na C-64

LOAD i SAVE

Po małej przerwie w poprzednim numerze KEBAB'a, dzisiaj znowu porcja informacji dla miłośników asemblera na C-64. Dzisiaj zajmiemy się podstawowymi operacjami I/O (we/wy) jakimi są funkcje LOAD i SAVE. Wszyscy wiemy jak prosto wykonuje się te operacje z poziomu BASIC'a.

Piszemy po prostu:

```
LOAD"NAZWAZBIORU",dev,sa
```

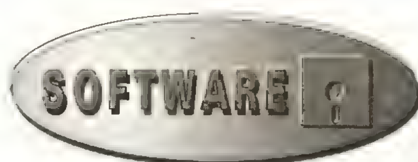
- gdzie "dev" oznacza tzw. numer urządzenia a "sa" to specjalny "adres wtórny" (ang.: secondary address).

Zapewne jest nam już wiadome, że w efekcie końcowym BASIC i tak poprosi o wykonanie zadania, procedury systemu operacyjnego. Sięgamy zatem po naszą tabelę skoków KERNAL'a (KEBAB nr 5/92) i szukamy jakichś znajomo brzmiących nazw.

Natychmiast widać pod adresami $FFD5 i $FFD8 dwa hasła identyczne z cytowanymi przed momentem funkcjami (komendami) BASIC'a. Mam tu na myśli oczywiście procedury LOAD i SAVE. Co zatem zrobić aby móc z poziomu języka asemblera skorzystać z w/w procedur? Ktoś mógłby powiedzieć, że to oczywiste: wykonujemy prosty skok pod wskazany adres np.:

```
JSR $FFD5
```

i mamy sprawę z głowy! Niestety tak nie jest. Co prawda będziemy



musieli wykonać JSR (lub JMP) $FFD5 aby załadować coś do pamięci, ale zanim to zrobimy to nie możemy zapomnieć o kilku istotnych sprawach. Po pierwsze przypomnijmy sobie jakie argumenty (parametry) podawaliśmy wywołując funkcję LOAD z poziomu BASIC'a.

1. Nazwa zbioru

2. Numer urządzenia z którego chcemy coś załadować

3. Adres wtórny

Otóż okazuje się, że jeżeli funkcję LOAD wywołamy bezpośrednio z systemu operacyjnego przy użyciu języka maszynowego to i tak musimy w jakiś sposób podać wszystkie wymienione parametry. Zaraz! A z magnetofonu?

Można napisać tylko LOAD i nic więcej... Owszem, można!

Ale tylko dlatego, że w standardowym systemie operacyjnym C-64, magnetofon jest tzw. urządzeniem domyślnym. Oznacza to, że jeżeli nie podamy numeru urządzenia, to system domyśli się, że chodziło nam o magnetofon (urządzenie numer jeden) i taką właśnie wartość przekaże wywołując funkcję LOAD.

Również nazwa zbioru nie jest niezbędna w przypadku ładowania z magnetofonu, gdyż system domyśli się, że chcemy załadować z taśmy pierwszy napotkany zbiór. Pozostaje jeszcze adres wtórny. On również ma swoją wartość domyślną. Jest nią zero. Jednakże system "domyśli się" tylko w przypadku gdy BASIC podpowie mu wszystko co trzeba. Jeżeli natomiast nie korzystamy z BASIC'a to musimy wszystkie wartości przekazać do systemu własnoręcznie i żadnej nie pominąć. Jak zatem tego dokonać? W tabeli skoków znajdują się trochę wyżej pod adresami $FFBA i $FFBD dwie procedury o nazwach SETLFS i SETNAM.

SETLFS to skrót od angielskiego (SET) (L)ogical (F)ile and (S)econdary address. SETNAM to skrót od (SET) (NAM)e.

Pierwsza pozwala nam na ustalenie kanału logicznego, numeru urządzenia oraz adresu wtórnego. Druga z kolei umożliwia podanie nazwy zbioru. Zacznijmy może od pierwszej. Załadujmy zatem akumulator wartością równą numerowi kanału logicznego, z którego chcemy skorzystać. W zasadzie numer ten nie ma specjalnego znaczenia i możemy dobrać go dosyć swobodnie. Musimy jednak pamiętać aby nie otwierać dwóch kanałów o tym samym numerze. Koderzy bardzo często stosują numerację kanału zgodną z numerem urządzenia lub z adresem wtórnym. Zróbmy i my podobnie. Napiszmy:

```
LDA #$08
```

Następnie musimy podać numer urządzenia, z którego będziemy ładowali nasz zbiór. Numer ten przekazujemy w rejestrze X. Zakładamy, że chcemy ładować ze stacji dysków o numerze osiem. Piszemy zatem:

```
LDX #$08
```

I pozostał nam jeszcze adres wtórny. Tu mamy kilka możliwości. Wartość zero oznacza, że zbiór nasz zostanie załadowany pod adres, który podamy w momencie wywoływania samej procedury LOAD.

Wartość różna od zera oznacza, że adres, pod który zbiór ma zostać załadowany, system ma pobrać z dwóch pierwszych bajtów ładowanego zbioru. Pamiętamy? W BASIC'u było podobnie. Jeżeli chcieliśmy załadować coś pod adres inny niż $0801, to musieliśmy napisać np.:

```
LOAD MONITOR ,8,1
```

Gdy natomiast pominęliśmy adres wtórny, to BASIC "domyślał się", że chcemy załadować jakiś zbiór na początek pamięci przeznaczonej dla programów w BASIC'u (standardowo $0801). W związku z tym "podpowiadał" systemowi, że chodzi o adres wtórny równy zero i adres ładowania równy aktualnemu początkowi pamięci BASIC'a.

My jednak chcemy pobrać adres ładowania bezpośrednio ze zbioru i robimy to ładując rejestr Y wartością adresu wtórnego różną od zera.

```
LDY #$01
```

Następnie wywołujemy z tabeli funkcję SETLFS skokiem

```
JSR $FFBA
```

i system już wie, że będziemy chcieli coś robić z urządzeniem numer osiem (stacja dysków), dokonamy tego poprzez ósmy kanał logiczny oraz adres wtórny dla naszych operacji wynosi jeden. Pozostaje nam jeszcze przekazać parametry dotyczące nazwy zbioru. Na sam początek musimy gdzieś w dostępnej pamięci umieścić tekst odpowiadający naszej nazwie.

Przykładowo przyjmijmy, że zbiór nasz będzie się nazywał "monitor". Umieścimy zatem w/w ciąg znaków w jakimś bezpiecznym miejscu np. pod adresem $7000. Następną czynnością będzie przekazanie systemowi informacji o tym gdzie znajdują się bajty tworzące żądaną nazwę zbioru. Do tego celu służy funkcja SETNAM.

Zanim jednak ją wywołamy, musimy jeszcze przekazać do niej kilka parametrów...

```
LDX #$00

LDY #$70
```

... czy domyślamy się już o co chodzi? Otóż adres $7000 rozłożyliśmy na dwa bajty (młodszy i starszy, LO/HI). Młodszy wpisujemy do rejestru X, starszy do rejestru Y. Będzie to oznaczało, że system ma szukać bajtów oznaczających nazwę zbioru począwszy od tego właśnie adresu.

Pozostało jeszcze tylko powiedzieć mu jak dużo tych bajtów składa się na naszą nazwę. Robimy to wpisując do akumulatora wartość równą liczbie znaków nazwy. W naszym przypadku jest to liczba siedem ($07).

```
LDA #$07
```

Wykonujemy skok do procedury SETNAM...

```
JSR $FFBD
```

... i mamy parametry przygotowane. Teraz pozostaje nam już tylko wykonać właściwe ładowanie zbioru. Robimy to następująco:

```
LDA #$00

JSR $FFD5
```

Wszystko jasne? Pod adresem $FFD5 znajduje się w tabeli skoków procedura LOAD. Po co jednak jeszcze "LDA #$00"? Otóż okazuje się, że ta sama procedura LOAD, może służyć zarówno do ładowania do pamięci jak i do weryfikacji poprawności zapisu. Pamiętamy z BASIC'a komendę "VERIFY"? Otóż załadowanie akumulatora wartością $00 oznacza, że mamy zamiar wykonać ładowanie. Wpisanie natomiast do akumulatora wartości $01 oznacza, że procedura "LOAD" ma po wywołaniu dokonać weryfikacji zapisu czyli po prostu porównania zawartości zbioru zapisanego na dyskietce (lub kasecie) z zawartością odpowiedniego obszaru pamięci komputera.

Jeżeli wszystko zrobiliśmy jak należy i na dyskietce znajdował się zbiór o podanej przez nas nazwie to powinien on zostać po wykonaniu ostatniego "JSR", załadowany do pamięci.

Prześledźmy zatem jeszcze raz od początku wszystkie kroki jakie musimy wykonać aby załadować jakiś zbiór.

1. Ustalić numer kanału logicznego, urządzenia oraz adres wtórny

a) numer kanału -> A
b) numer urządzenia -> X
c) adres wtórny -> Y
d) skok do procedury SETLFS ($FFBA)

2. Ustalić nazwę zbioru

a) umieszczamy nazwę zbioru w dostępnej pamięci RAM, posługując się kodami PETASCII (tablice kodów znajdują się na końcu instrukcji obsługi)
b) długość nazwy -> A
c) młodszy bajt adresu nazwy -> X
d) starszy bajt adresu nazwy -> Y
e) skok do procedury SETNAM ($FFBD)

3. Wywołać procedurę LOAD

a) ustalamy rodzaj pracy procedury "LOAD" (ładowanie/weryfikacja)
b) w przypadku adresu wtórnego różnego od zera podajemy adres ładowania
- młodszy bajt -> X
- starszy bajt -> Y
c) skok do procedury LOAD ($FFD5)

Na początek może się to wszystko wydawać skomplikowane ale jak zrozumiemy reguły, to okaże się, że będziemy wykonywali wszystkie kroki automatycznie w naszych programach.

W przypadku korzystania z magnetofonu możemy sobie sprawę znacznie uprościć gdyż nie musimy podawać nazwy zbioru. Wystarczy przed wywołaniem procedury SETNAM podać długość nazwy równą zero (wpisując $00 do akumulatora) i nie musimy się już martwić więcej o nazwę.

Musimy natomiast pamiętać jeszcze o tym, że magnetofon ma numer urządzenia równy jeden i taką właśnie wartość musimy wpisać do rejestru X przed skokiem pod $FFBA.

Tyle na temat ładowania. Teraz zajmijmy się transmisją w drugą stronę. Załóżmy, że chcemy zapisać jakiś zbiór na dyskietce. Do tego celu służy procedura SAVE. Zapisuje ona kopię zawartości pewnego obszaru pamięci komputera na jakimś zewnętrznym nośniku informacji np. dyskietce. Musimy jednak zaznaczyć JAKI obszar pamięci zamierzamy skopiować (zapisać).

O ile w przypadku BASIC'a komenda SAVE działała bez dodatkowych parametrów dotyczących adresów początku i końca obszaru pamięci przeznaczonego do zapisania o tyle w przypadku bezpośrednich odwołań do KERNAL'owskiej procedury SAVE musimy przekazać dodatkowo w/w adresy.

Jak to zrobić powiemy za chwilę. Zanim bowiem zabierzemy się za właściwe zapisywanie na dysku lub taśmie, musimy podobnie jak przy ładowaniu, ustalić wszystkie parametry dotyczące kanału logicznego, nazwy itd.

Aby się nad tym zbytnio nie rozwodzić powiem tylko, że robi się to analogicznie jak w przypadku ładowania. Ustalamy nazwę, kanał logiczny, numer urządzenia oraz adres wtórny (równy jeden) korzystając z omówionych przed chwilą procedur SETLFS i SETNAM. Gdy już nam się to uda, możemy przystąpić do końcowego etapu operacji SAVE.

Tu właśnie okazuje się niezbędne podanie adresów, o których wspominałem powyżej. Adres końcowy obszaru przeznaczonego do zgrywania podajemy w rejestrach X i Y wpisując zwyczajowo (przed wywołaniem procedury SAVE) młodszy bajt adresu do rejestru X, a starszy do rejestru Y. Pozostaje problem z adresem początkowym gdyż mamy do dyspozycji już tylko jeden rejestr - akumulator, a potrzebujemy minimum dwóch bajtów na zapisanie dowolnego adresu z szesnastobitowej przestrzeni adresowej C-64.

Jak zatem sobie poradzić? Otóż procesor 6502 oraz jego odmiana 6510 zawarta w każdym "komciu", posiada specjalnie wyodrębnione tryby adresowania pierwszych 256 komórek z obszaru adresowego czyli tzw. strony zerowej (ang.: zeropage).

Dzięki tym specjalnym trybom adresowania, wszelkie operacje przeprowadzane na tych komórkach mogą być wykonywane z prędkością porównywalną do prędkości operacji na rejestrach procesora. W związku z tym system operacyjny wykorzystuje większość w/w komórek do swoich potrzeb. Między innymi cała procedura służąca do odczytu tekstu programu w BASIC'u jest po każdym resecie umieszczana na stronie zerowej ($73) aby działała szybciej.

My jednak wróćmy do naszych problemów z SAVE i z adresem początkowym obszaru pamięci przeznaczonego do zapisu na dysku/taśmie. Otóż okazuje się, że twórcy systemu byli na tyle uprzejmi, że pozostawili na stronie zerowej kilka komórek nie wykorzystanych. Przykładowo od adresu $FB znajdują się cztery takie komórki (do $FE włącznie). Przyjmijmy zatem, że do komórki $FB wpiszemy młodszy bajt adresu początku obszaru.... a do komórki $FC, starszy. Sprawa zatem przedstawia się następująco:

START młodszy -> $FB
START starszy -> $FC
END młodszy -> X
END starszy -> Y

Na koniec jeszcze tylko informujemy system o tym gdzie (na stronie zerowej) umieściliśmy nasze wskaźniki, wpisując numer komórki strony zerowej, która zawiera młodszy bajt adresu, do akumulatora.

```
LDA #$FB
```

i wykonujemy skok do procedury SAVE:

```
JSR $FFD8
```

Czy to jest jasne? Z pewnością jeszcze nie! Ale kilka eksperymentów powinno wyjaśnić wątpliwości. Do następnego spotkania zatem.

SD!

Mapa pamięci



0E0 BPL1PTH Z + A
0E2 BPL1PTL Z + A
...
0F4 BPL6PTH Z + A 0F6 BPL6PTL Z + A

Wskaźniki Bit-Planeów. Do rejestrów tych wpisujemy adres od którego chcemy wyświetlić pamięć. Adres ten powinien być parzysty i wskazywać pamięć typu CHIP.

100 BPLCON0 Z AD
102 BPLCON1 Z D 1
04 BPLCON2 Z D

Rejestry kontroli wyświetlanego obrazu. Przy ich pomocy definiujemy rozdzielczość ekranu, ilość kolo-

rów, itp. Oto jak wygląda znaczenie poszczególnych bitów:

BPLCON0:

bit 15 (HIRES) - Ustawiony włącza tryb HIRes.

bity 14 - 12 (BPU2-0) - Ta trójka bitów może przyjmować wartości z przedziału od 0 do 6 (czyli od %000 do %110). Wartość różna od zera i sześciu określa ilość włączonych Bit-Planeów, tym samym określając ilość kolorów następującą zależnością:

ilość_kolorów = 2 ^ wartość_tu_wpisana

Wartość 6 używana jest w trybie EHB lub HAM.

bit 11 (HOMOD) - Ustawiony włącza tryb HAM (Hold and Modify), skasowany przy włączonych 6 Bit-Planeach oznacza tryb EHB (Extra Half Bright).

bit 10 (DBLPF) - Włącza tryb Double Playfield.

bit 9 (COLOR) - Używany w Amidze 1000.

bit 8 (GAUD) - Używany przy współpracy z Genlockiem.

bity 7 - 4 - Nie używane.

bit 3 (LPEN) - Umożliwia współpracę z piórem świetlnym.

bit 2 (LACE) - Włącza tryb Interlace.

bit 1 (ERSY) - Używany we współpracy z Genlockiem.

bit 0 - Nie używany.

BPLCON1:

bity 15 - 8 - Nie używane.

bity 7 - 4 (PF2H3-0) - Określają wartość sprzętowego przesunięcia parzystych Bit-Plane'ów.

bity 3 - 0 (PF1H3-0) - Określają wartość sprzętowego przesunięcia nieparzystych Bit-Plane'ów.

BPLCON2:

bity 15 - 7 - Nie używane.

bit 6 (PF2PRI) - Ustawiony oznacza, że Bit-Plane'y parzyste mają priorytet (będą pojawiać się z przodu) nad Bit-Plane'ami o numerach nieparzystych. Używane w trybie Dual-Playfield.

bity 5 - 3 (PF2P2-0) - Określają priorytet Playfield'u 2 w stosunku do Sprite'ów.

bity 2 - 0 (PF1P2-0) - Określają priorytet Playfield'u 2 w stosunku do Sprite'ów.

108 BPL1MOD Z A
10A BPL2MOD Z A

Rejestry modulacji ekranu. Określają wartość jaka będzie dodana do wskaźnika pamięci obrazu po wyświetleniu każdej linii. Dzięki temu część pamięci ekranu nie jest wyświetlana na ekranie.

Zanim przejdziemy do szczegółowego opisu techniki tworzenia kolorowego obrazu proszę Czytelnika o zapoznanie się z treścią odcinka tego cyklu zamieszczoną w Kebabie nr 6. W tamtej części rozpatrywaliśmy przypadek zorganizowania pamięci dla wyświetlenia obrazu jednokolorowego. Mieliśmy wtedy ustawiony tylko jeden Bit-Plane, czyli tzw. bitową mapę kolorów. Bit ustawiony wskazywał procesorowi graficznemu, że ma narysować punkt w kolorze pobranym z rejestru pierwszego, natomiast bit skasowany na ekranie był odzwierciedlany punkcikiem w kolorze o kodzie pobranym z rejestru zerowego. Zastanówmy się teraz jak będzie w przypadku ustawienia 2 Bit-Plane'ów (wpisujemy %010 na miejsce bitów 14-12 rejestru BPLCON0). W takiej sytuacji każdy punkt na ekranie będzie określany przy pomocy 2 bitów w pamięci - po jednym bicie na każdym Bit-Plane'ie. Owe dwa bity mogą przyjmować następujące wartości:

1. Oba skasowane. (%00 = dec 0) **2.** Pierwszy ustawiony, drugi skasowany. (%01 = dec 1) **3.** Pierwszy skasowany, drugi ustawiony. (%10 = dec 2) **4.** Oba ustawione. (%11 = dec 3)

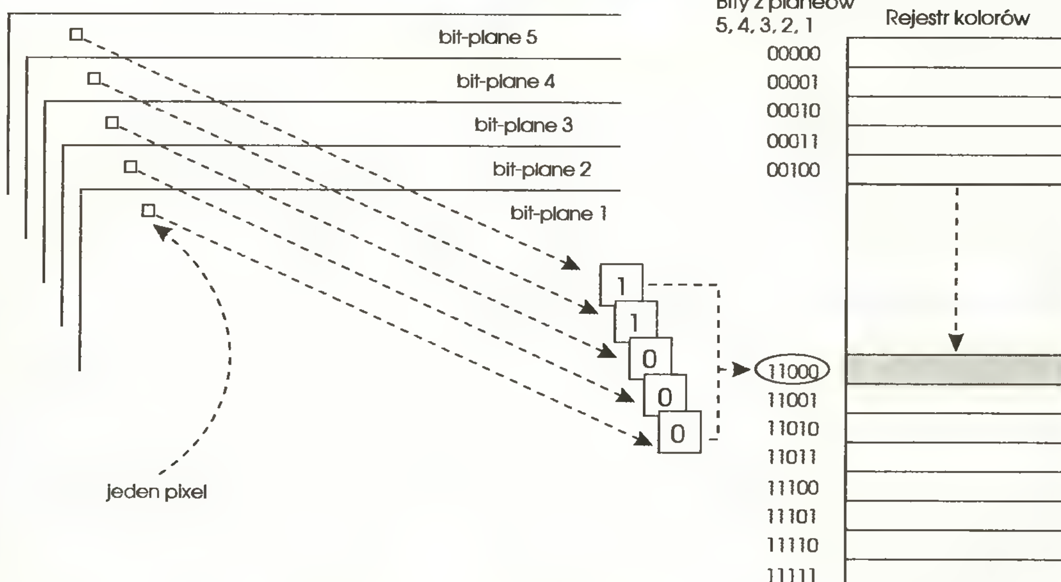
Proszę zauważyć, że występują tu cztery możliwe kombinacje. Każda z nich wskazuje numer REJESTRU KOLORU z którego ma zostać pobrany kod koloru w celu odzwierciedlenia zawartości pamięci obrazu na ekranie. Wynika z tego, że ilość wspomnianych kombinacji jest równa ilości dostępnych kolorów. Wiadomo już zatem dlaczego aby uzyskać 4 kolory na ekranie musimy włączyć dokładnie 2 Bit-Plane'y. Dla lepszego zrozumienia zagadnienia zamieszczamy na następnej stronie rysunek obrazujący definicję koloru jednego pixela na ekranie przy włączonych 5 Bit-Plane'ach.

Jak już wiemy odpowiednia kombinacja bitów pobranych z kolejnych Bit-Plane'ów określa nam numer rejestru koloru.

Rejestry te nazywają się
COLOR0,
COLOR1,
COLOR2,... ,
COLOR31
i posiadają adresy $0180, $0182, $0184, itd... Poszczególne bity każdego z nich oznakowane są następująco:

bity 15-12 - nie używane
bity 11-8 (R3-R0) - Składowa czerwona.
bity 7-4 (G3-G0) - Składowa zielona.
bity 3-0 (B3-B0) - Składowa niebieska.

i dopiero procesor graficzny bazując na danych pobranych z tych rejestrów kreśli pixel w odpowiednim kolorze. Analogii z takim mechanizmem tworzenia obrazu można szukać wśród trybów adresowania procesora, a konkretnie z trybem pośrednim rejestrem adresowym. Załóżmy, że nasze bity to rejestr adresowy procesora. One wskazują rejestr koloru, tak jak rejestr adresowy wskazuje jakiś adres. Zawartość wskazywanego rejestru koloru zostaje pobrana i naniesiona na ekran (co my widzimy w postaci barwnego punktu na monitorze), tak jak procesor do operandu przeznaczenia "wrzuci" to co jest jest wskazywane przez adres znajdujący się w rejestrze adresowym.



Nieco bardziej rozszerzony mechanizm tworzenia obrazu występuje w trybie EHB (Extra Half Bright - dodatkowa połowa jasności). Jak wiemy tryb ten dostarcza nam możliwość wyświetlenia jednocześnie na ekranie 64 kolorów. Dla ich zdefiniowania w pamięci potrzebnych jest 6 BitPlane'ów (2^6=64), oraz zgodnie z tym co powiedzieliśmy wcześniej: 64 rejestry kolorów.

Jednakże Amiga tych ostatnich posiada jedynie 32. Okazuje się, że wprowadzenie 64-kolorowego trybu EHB było możliwe dzięki zastosowaniu pewnego tricku. Otóż wszystkie pixele opisane w pamięci ekranu, posiadające zapalony bit na 6 BitPlanie (czyli niejako odwołujące się do nieistniejących rejestrów 32-63) zostaną wyświetlone na ekranie w kolorze pobranym z rejestru o numerze pomniejszonym o 32, ale ze zmniejszoną o połowę jasnością.

Zatem nasza Amiga w pełni niezależnych od siebie kolorów może wyświetlić jednocześnie na ekranie tylko 32. W trybie EHB pozostałe 32 są tymi samymymi kolorami, tylko o połowę ciemniejszymi. Doskonałym przykładem tego mechanizmu jest okienko definiowania kolorów w DPaincie. Proszę sobie otworzyć ekran główny tegoż programu w trybie EHB, i korzystając ze wspomnianego okna spróbować zmienić wartość któregoś z kolorów - i choć modyfikować będziemy tylko jeden, na ekranie zmnieniać się będą dwie barwy.

Krzysztof Kobus

JAK ZROBIĆ WŁASNY TURBOLOADER czyli programowanie stacji 1541/71 część 3.



Ostatnim razem omówiliśmy różne metody przesyłania bajtów swojego programu do pamięci RAM stacji dysków. Omówiliśmy także kilka przykładowych programów.

Sądzę więc, że nie powinno to już być dla nikogo problemem. Posuńmy się więc o jeden krok dalej. Co musimy teraz zrobić? Oczywiście, uruchomić nasz program w RAM stacji. Bardziej "fachowo" możnaby powiedzieć: przejąć kontrolę nad stacją przez nasz własny program, czy jakoś inaczej, ale nie to jest istotne.

Postawmy więc pytanie: jakie mamy możliwości techniczne w tym względzie? Okazuje się, że niezbyt wielkie. Istnieje praktycznie jeden rozkaz służący do tego celu, który jest na szczęście dość wygodny w użyciu. Jest nim MEMORY-EXECUTE (w skrócie M-E). Jego składnia wygląda następująco:

```
"M-E"+CHR$(lo)+CHR$(hi)
```

Tak, tak, jest to kolejny rozkaz z grupy rozkazów wewnętrznych (internal commands). Czy więc trzeba więcej o nim mówić? Poprzez analogię do innych rozkazów z tej grupy łatwo się domyśleć, co oznaczają bajty "lo", oraz "hi". Oczywiście jest to młodszy i starszy bajt adresu w stacji, pod którym znajduje się nasza procedura do uruchomienia. Można powiedzieć, że jest to prawie dokładnie odpowiednik polecenia SYS z BASIC-a, czy JSR z asemblera, tyle że działa wewnątrz stacji, a nie komputera.

Nie trzeba chyba nikomu wyjaśniać, że rozkaz ten posyłać należy przez kanał rozkazowy (kanał 15.) W tym miejscu powinien znaleźć się przykład wykorzystujący MEMORY-EXECUTE dla uruchomienia jakiegoś ciekawego efektu w stacji. Dlaczego jednak tego nie robię?

Otóż okazuje się, że o ile w przypadku komputera SYS zademonstrować można na tysiąc różnych sposobów, o tyle w przypadku stacji nie jest to takie proste. Wszystko przez to, że ROM stacji 1541 jak i 1571 napisany jest bardzo nieelegancko. Jest tam bardzo niewiele procedur, które nadają się do wykorzystania przez użytkownika, którego niekoniecznie interesuje obsługa stacji wyłącznie przez jej system operacyjny (DOS), a który chciałby zaprogramować swoje własne, lepsze procedury. Owo "nienadawanie się" procedur do użytku polega na tym, że zostały one napisane jako części składowe całego systemu i nie zostało przewidziane ich używanie oddzielne.

Problemem jest przeważnie to, że praktycznie wszystkie procedury, które mogą stwierdzić jakikolwiek błąd dyskowy (a więc wszystkie, robiące coś sensownego) skaczą w takim wypadku do odpowiedniego miejsca systemu operacyjnego, w którym zamieszczona jest standardowa obsługa błędów.

Wygląda to w ten sposób, że wywołujemy taką procedurę przez JSR (z programu wykonywanego przez procesor stacji oczywiście) i w wypadku, gdy wszystko jest OK, procedura "grzecznie" wraca poprzez RTS. Gdy jednak coś jest nie tak, na przykład znaleziono jakieś błędy dyskowe, czy też użytkownik "zapomniał" włożyć dysk do stacji, to procedura ta skacze do głównej obsługi błędów DOS poprzez JMP i w tym momencie tracimy kontrolę nad stacją.

Nasz program może ponownie zacząć działać po wysłaniu kolejnego M-E. Czy jednak konieczne jest pisanie wszystkiego od zera? Aż tak źle nie ma. Jak się niedługo okaże, część systemu operacyjnego można wykorzystać, pod warunkiem jednak, że wie się dokładnie jak to wszystko działa.

Powiemy o tym więcej już wkrótce. Wróćmy jednak do rzeczywistości i zajmijmy się naszym MEMORY-EXECUTE.

Powiedziałem, że jest to bardzo wygodny rozkaz. Co jednak jest tego przyczyną? Zastanówmy się jak wygląda posyłanie tego rozkazu do stacji. Na początku wywołujemy LISTEN i SECOND, aby stacja o interesującym nas numerze przyłączyła się do szyny szeregowej. Następnie wysyłamy 5 bajtów rozkazowych (M-E,lo,hi) używając CIOUT. A co w tym czasie robi stacja? Jej jedyne zajęcie, to zapewnienie poprawnej komunikacji przez szynę szeregową. Nas interesuje jedno: co dzieje się z kolejno przychodzącymi bajtami. Wiemy już z poprzednich odcinków, że bufor wejściowy zaczyna się od adresu $200 i tam właśnie magazynowane są te bajty. Wiemy też, że bufor ten ma długość 41 bajtów. Co więc się stanie, jeżeli poślemy do stacji następującą sekwencję:

```
"M-E"+CHR$(5)+CHR$(2)+
CHR$(bajt1)+CHR$(bajt2)+...
```

Odpowiedź nasuwa się sama. Nasz ciąg bajtów znajdzie się w stacji w buforze wejściowym, a po wykonaniu UNLISTEN stacja skoczy pod podany adres. W tym wypadku jest to $205, czyli adres, gdzie zaczynają się bajty posłane jakby jako dodatkowe parametry MEMORY-EXECUTE. Jaki z tego wszystkiego możemy wyciągnąć wniosek? Dla nas najistotniejsze jest to, że jednym rozkazem jesteśmy w stanie wysłać i uruchomić krótki programik (nie dłuższy niż 36 bajtów), zawierający na przykład kod dla procesora stacji do obsługi szybkiego przesłania reszty programu, czy też coś innego. Przykłady zastosowań można mnożyć. Spróbujmy teraz zastosować nasze wiadomości w praktyce. Na początek wpiszmy w BASIC-u taki program:

```
10 OPEN 1,8,15
20 PRINT#1,"M-E"CHR$(4)CHR$(2)
```

i spróbujmy go wykonać. No i co? Żadnego rezultatu? A może jednak. Spróbujmy wysłać teraz jakiś rozkaz do stacji. Wpiszmy na przykład `LOAD"$",8`.

No i co? Co się dzieje z naszą stacją? Bez obawy, sprzętowi nic się nie stało.

Spowodowaliśmy jedynie kompletne zablokowanie stacji, każąc jej procesorowi wykonać rozkaz o kodzie $02, znajdujący się pod adresem $204 jako starszy bajt właśnie tego adresu. Jedyny sposób, aby stację odblokować, to zresetowanie urządzenia (chodzi oczywiście o reset sprzętowy, a nie programowy). Tym, którzy nie pamiętają jaki rozkaz kryje się pod kodem $02 proponuję dokładniejsze przestudiowanie listy rozkazów "niepublikowanych" procesora 6502 i jemu podobnych. Małego usprawiedliwienia wymaga brak zamknięcia zbioru po zakończeniu programu. Jeżeli kogoś interesują powody, niech spróbuje dopisać linię:

```
30 CLOSE 1
```

i uruchomić taki program. Gdy to zrobimy, program ten nigdy się nie skończy, gdyż zatrzyma się na CLOSE oczekując na sygnały ze stacji, których ta nie jest w stanie wygenerować.

Jak już wiemy z poprzednich odcinków CLOSE nie jest konieczne, aby poprawnie przesłać i wykonać rozkaz dyskowy. Z naszego testu wynika jednak, że rozkaz ten COŚ robi ze stacją. Tak też jest w istocie.

To COŚ jednak w naszym wypadku nie jest do niczego potrzebne, a wynika z uniwersalności BASIC-owego (a raczej KERNAL-owego) CLOSE, które oprócz zarządzania wskaźnikami dotyczącymi zbiorów, powoduje też zamknięcie pliku otwartego do odczytu, czy (co ważniejsze) do zapisu. Jak łatwo się domyśleć zamknię-

cie to polega z punktu widzenia komputera na wysłaniu specjalnej sekwencji kodów przez port szeregowy, co w naszym przypadku spowodowało zablokowanie komputera.

Nie znaczy to jednak, że komputer się zawiesza. Gdy wgłębić się w specyfikę działania SERIAL-PORT-u, to okaże się, że komputer wysyła sygnał, że chce coś przesłać, na co elektronika stacji (NIE procesor!) odpowiada, że komputer ma czekać, bo gdy tylko procesor będzie miał "wolną chwilę" to zajmie się transmisją.

Ponieważ procesor po napotkaniu kodu $02 nie wykona już żadnego rozkazu, aż do jego wyzerowania przez reset, logiczne jest, że komputer czeka w nieskończoność. Brak CLOSE nie wygląda może najlepiej, ale z drugiej strony po co komu używać jakichś zbiorów, jeżeli stacja "wisi"? Zostawmy już ten przykład w spokoju i wpiszmy coś innego (tym razem już w asemblerze):

```
 *= $2000
          lda #$08
          jsr $ffb1
          lda #$6f
          jsr $ff93
          ldx #$00
loop1     lda command,x
          jsr $ffa8
          inx
          cpx #27
          bne loop1
          jmp $ffae
command   .byte $4d,$2d,$45,$05,$02
          sei
loop3     lda $1c00
          eor #$08
          sta $1c00
          ldy #$00
          ldx #$00
loop2     nop
          dex
          bne loop2
          dey
          bne loop2
          beq loop3
```

Jak łatwo zauważyć, program ten wysyła do stacji 27 bajtów, z których pierwsze 5 to rozkaz MEMORY-EXECUTE, powodujący uruchomienie programu rozpoczynającego się od następnego bajtu. Nie zagłębiając się w szczegóły powiem tylko, że bit 3 (o wartości 8) w rejestrze $1c00 steruje diodą stacji opisaną na obudowie jako DRIVE.

Jeżeli zapalimy ten bit, to dioda się świeci, a gdy go zgasimy, to gaśnie. Częstotliwość migotania diody regulować możemy, zmieniając opóźnienie wnoszone przez pętlę opóźniającą zrealizowaną z użyciem rejestrów .X i .Y. Należy tylko pamiętać o ewentualnym zmodyfikowaniu pętli wysyłającej bajty do stacji, jeżeli zmiana opóźnienia spowodowała wydłużenie programu.

Jeżeli nasze eksperymenty przeprowadzamy ze stacją 1571, to możemy wizualnie przekonać się jaki wpływ ma zmiana częstotliwości zegara systemowego stacji na prędkość wykonywania programu.

Proponuję spróbować najpierw, jak program taki działa ze stacją przełączoną w tryb 1541 (np. rozkazem U0>M0) z zegarem 1 MHz, a następnie ze stacją w trybie 1571 (włączonym np. przez U0>M1) z zegarem 2 MHz.

Nietrudno zauważyć różnicę. Za każdym razem, gdy chcemy przerwać działanie naszego programu konieczne jest resetowanie stacji, ponieważ program ten nigdy się nie skończy i stacja staje się przez to praktycznie niedostępna. Rzeczą wartą zauważenia jest zastosowanie rozkazu BEQ w celu wykonania skoku na początek.

Wstawienie w to miejsce JMP loop3 spowodowałoby, że asembler (trochę dwuznaczne to określenie; chodzi mi teraz o program służący do pisania i kompilacji programu maszynowego) w czasie kompilacji wstawiłby w to miejsce wyliczony adres, pod jakim znajduje się ten program w komputerze jako kod wynikowy kompilacji, który bynajmniej nie byłby zgodny z adresem do którego należy skoczyć.

W naszym wypadku nie ma jednak problemu. Wstawiamy BEQ i po kłopocie. Nie zawsze jednak takie postępowanie będzie możliwe. Możemy przecież czasem potrzebować skoczyć dalej niż pół strony pamięci w przód, czy w tył. Możemy też potrzebować zmodyfikować program w czasie jego działania, czy też odwołać się do jakiejś tablicy wchodzącej w skład programu. W takim wypadku nie będziemy w stanie (jeżeli używamy TURBO-ASSEMBLER-a V5.1, czy jakiegoś podobnego) umieszczać zarówno procedur dla stacji, jak i dla komputera w jednym programie źródłowym. Najprostsze rozwiązanie to napisać wersję źródłową programu dla stacji, jako adres początkowy podając na przykład $300 (czyli adres, gdzie program będzie faktycznie wykonywany), skompilować ją na dysk (strzałka w lewo i "5" w TASS V5.1), skasować program źródłowy, po czym wpisać, lub wczytać z dysku źródłówkę programu dla komputera. Następnie należy wgrać skompilowany kod dla stacji jako dane pod adres np. $2000 (strzałka, shift+L), po czym za pomocą kombinacji strzałka, "6" spowodować wstawienie wczytanych bajtów jako linie .BYTE do programu źródłowego kodu dla komputera. Od tej pory mamy dostępne obydwa programy w jednym pliku źródłowym. Jedynym mankamentem takiego rozwiązania sprawy jest niemożność modyfikacji programu dla stacji. Gdy potrzebujemy wprowadzić w nim jakieś poprawki, to całą procedurę musimy zacząć od początku. Zalecam w związku z tym poświęcić nieco więcej uwagi programowi dla stacji, aby było w nim jak najmniej błędów, co pozwoli zaoszczędzić nam sporo czasu i niepotrzebnej pracy. To chyba by było tyle na temat uruchamiania programu przez MEMORY-EXECUTE. Oprócz tego istnieją inne, aczkolwiek bardzo rzadko używane sposoby uruchamiania. Jako przykład podać można BLOCK-EXECUTE. Rozkaz ten ma następującą składnię:

```
"B-E:channel,drive,track,sector"
```

Umożliwia on odczytanie z dysku wybranego przez nas sektora do bufora, który jest przydzielony podanemu kanałowi, a następnie skok do pierwszego bajtu tego

bloku. (Jest to praktycznie takie połączenie BLOCK-READ i MEMORY-EXECUTE.) Już na pierwszy rzut oka widać podstawową wadę tego rozkazu. Przed posłaniem go (przez kanał 15 oczywiście) konieczne jest otwarcie kanału bezpośredniego dostępu np. oddzielnym OPEN.

Poza tym wszystkie wartości liczbowe muszą być podane jako kody ASCII. Na przykład, aby wysłać numer ścieżki (parametr "track") równy dajmy na to 18, musimy wysłać bajty: $31,$38. Gdy któryś z parametrów ma się zmieniać, to musimy dodatkowo pisać procedurę konwersji liczby na tekst. Nie będę się już rozwodził na temat niekopiowalności zbiorami programu ładującego program dla stacji bezpośrednio z któregoś sektora.

Jak widać: same problemy z tym rozkazem. Wszystko to spowodowało, że rozkaz ten jest praktycznie nie używany. W większości przypadków całkowicie wystarcza MEMORY-EXECUTE.

Spróbujmy teraz napisać jakiś program dla stacji, który robiłby coś, co możnaby wykorzystać pisząc turboloader. Wszystko pięknie, ale w którym miejscu w stacji go umieścić? Wiemy już, że mamy do dyspozycji 2 kB pamięci RAM dostępne na samym początku przestrzeni adresowej procesora (czyli od $0000 do $07ff).

Jest jednak oczywiste, że część pamięci musi być zarezerwowana do użytku przez system operacyjny i umieszczenie tam swojego programu wiązało się będzie z wieloma niepotrzebnymi komplikacjami. Omówmy więc w skrócie które obszary do czego są wykorzystywane:

$0000-$00ff - strona zerowa procesora. Jest to praktycznie w całości obszar roboczy DOS-u. Przechowywanych jest tu także wiele informacji o stanie stacji, jak na przykład jaki stacja ma numer, czy dyskietka się kręci, jak długo ma się jeszcze kręcić, czy dioda ma świecić, jakie ostatnio wystąpiły błędy dyskowe itp.

Część lokacji na stronie zerowej jest wykorzystywane wewnątrz procedur do przechowywania aktualnych wartości, część jest używane tylko przez niektóre procedury i z tego powodu można je używać we własnych programach.

$0100-$01ff - stos systemowy procesora. Część tej strony to faktycznie stos systemowy. Ponieważ jednak okazało się, że na stos w zasadzie nie potrzeba aż 256 bajtów, a RAM-u w stacji nie ma zbyt dużo, postanowiono nie marnować pamięci i wykorzystać ten blok także do innych celów. Obszar pomiędzy $01bb i $01ff używany jest jako pamięć robocza w czasie odczytu i zapisu sektora na dysk. Dodatkowo, w obszarze od $0146 do $01af, stacja 1571 przechowuje BAM (mapę zajętych bloków) drugiej strony dysku. Stos systemowy zaczyna się więc od $0145 i w miarę jego używania rozbudowuje się w dół.

$0200-$02ff - dodatkowy obszar roboczy używany głównie przy pracy z różnymi plikami, kanałami, czy też z katalogiem i BAM-em dysku (np. w czasie zapisu pliku). W obszarze $0200-$0229 znajduje się ponadto bufor wejściowy używany w czasie pobierania ciągów bajtów rozkazowych przychodzących z komputera, a w obszarze $02d5-$02f8 bufor służący do generowania komunikatów o błędach wykrytych przez DOS.

$0300-$07ff - pięć buforów, służących do magazynowania aktualnie czytanego sektora, aktualnego bloku katalogu itp. W obszarze tym najwygodniej jest umieszczać kod swojego programu, gdyż dokładnie wiadomo, kiedy i w jakim celu DOS się do niego odwołuje.

$0800-$0fff - w obszarze tym znajduje się tzw. powtórzenie obszaru $0000-$07ff. Oznacza to, że chcąc np. pobrać wartość z komórki $0300 możemy tak samo dobrze napisać LDA $0300 jak i LDA $0b00.

Dzieje się tak z tego powodu, że układ zarządzający przestrzenią adresową stacji, w wypadku gdy adres jest mniejszy od $1000 przyjmuje, że procesor chce się odwołać do pamięci RAM. Pamięć ta zostaje więc uaktywniona, a jako że dochodzi do niej tylko 11 najmłodszych bitów słowa adresowego (2 kB zaadresować można za pomocą właśnie 11 bitów) nie ma znaczenia, do którego obszaru procesor faktycznie się odwołał. Pomimo istnienia możliwości utrudnienia komuś niepowołanemu analizy naszego programu, odradzam jednak pisania np. JMP $0f00, gdy chcemy skoczyć na $0700, ponieważ owo powtórzenie RAM-u nie jest oficjalnie określone przez firmę Commodore i nie na wszystkich stacjach musi być dostępne. Może się zdarzyć, że obszar ten nie będzie do niczego podłączony.

$1800-$180f - rejestry pierwszego układu VIA (VIA #1). Układ ten obsługuje transmisję przez port szeregowy, oraz (tylko w 1571) umożliwia sprawdzenie, czy głowica znajduje się nad ścieżką zerową, pozwala na przełączanie szybkiej szyny szeregowej 1571 pomiędzy wejściem, a wyjściem, na przełączanie aktywnej głowicy, oraz na przełączanie częstotliwości zegara systemowego pomiędzy 1 MHz i 2 MHz.

$1c00-$1c0f - rejestry drugiego układu VIA (VIA #2). Układ ten umożliwia sterowanie diodą LED, silnikiem napędzającym dyskietkę, silnikiem krokowym przesuwającym głowice, oraz zapisem bajtów na dyskietkę.

$2000-$2003 - rejestry układu WD 1770 (tylko w 1571). Układ ten zajmuje się odczytem i zapisem dyskietek w formacie MFM (stosowanym między innymi w stacjach komputerów zgodnych z IBM).

$4000-$400f - rejestry układu CIA (także tylko w 1571). Układ ten używany jest do sterowania szybką transmisją przez port szeregowy. Wszystkie te układy zostaną szczegółowo omówione w którymś z następnych odcinków.

$8000-$bfff - dodatkowa pamięć ROM 1571.

$c000-$ffff - pamięć ROM 1541/71.

Wszystkie obszary, które nie zostały wymienione, nie są do niczego podłączone i można się spodziewać, że na różnych stacjach będą się one różnie zachowywać. Wiemy już mniej więcej jak zorganizowana jest przestrzeń adresowa naszej stacji, możemy

więc zacząć pisać programy, które tam by się wykonywały. Prześlijmy pod adres $0400 w stacji coś takiego:

```
   *= $0400

        lda #$12
        sta $06
        lda #$01
        sta $07
        lda #$80
        sta $00
wait    lda $00
        bmi wait
        rts
```

Włóżmy następnie do stacji jakąś dyskietkę, zainicjujmy ją (np. rozkazem "I"), wykonajmy nasz program i sprawdźmy, co znajduje się od adresu $0300 (w stacji oczywiście). Jak łatwo zauważyć jest to początek katalogu naszego dysku. Jak to jest możliwe, że taki krótki program odczytał z dysku sektor?

Czyżby zadziałały tu przerwania? Jak się okazuje na 1541/71 w miarę łatwo da się odczytywać i zapisywać dysk, korzystając z usług przerwań i systemu operacyjnego. Inna sprawa, że prędkość nie zawsze nas zadowala. Na razie jednak interesuje nas bardziej wynik niż czas, a warto poznać, co DOS ma nam do zaoferowania.

Następnym razem zajmiemy się dokładniej tym mechanizmem. Zachęcam do eksperymentowania. Ostrzegam jednak, że przez nieuwagę można łatwo stracić dane z dysku na którym przeprowadzamy testy. Radzę więc brać dyski czyste, albo takie, których nie będziemy już potrzebować.

Krzysztof Matula

SOFTWARE

Amos VII

W tym miesiącu odpoczniemy od nauki, a zajmiemy się inną wersją Amosa o nazwie Easy Amos.

Program ten był szeroko reklamowany jako język, którego nauczyć się są w stanie nawet dzieci. Najbardziej wydatnie o tym świadczy miły "patron" tegoż programu - zabawny bobas z wąsami i literą "a" na koszulce. Musimy pamiętać, że w dziedzinie komputerów określenia takie jak m.in. przedszkolak mają zupełnie inne znaczenie (pamiętacie rubrykę "Tylko dla przedszkolaków" w Bajtku?). Także i tu bobas z wąsami nadaje podobny charakter programowi.

Tym razem, podobnie zresztą jak w przypadku "normalnego" Amosa, dostajemy do dyspozycji nie tylko sam interpreter, ale i garść programów wspomagania, lecz w tej wersji nowością jest pakiet uczący.

Oryginalnie mamy 2 dyskietki instalacyjne, z których rozpakowujemy nowego Amosa na trzy dyski. Tak oto otrzymujemy :

1. Easy_Amos, czyli dysk zawierający główny program wraz z innymi, niezbędnymi plikami np. bibliotekami.

2. Easy_Examples - zawiera przykładowe źródłówki (w tym gry).

3. Easy_Tutorial - z tego dysku uruchamiamy wspomniany już wcześniej pakiet uczący komend i technik Amosa, testujący również nabytą wiedzę.

Rozpoczniemy od najważniejszej części. czyli od edytora. Jest on w zasadzie podobny do pierwowzoru i kto pewnie poruszał się po starym Amosie, nie będzie miał najmniejszych problemów z posługiwaniem się nowym. To co od razu rzuca się w oczy to zmieniona grafika (na lepsze), oraz nieco inne obłożenie funkcji. Bezpośrednio z ekranu możemy wybrać opcję Tutor i Help.

Tutor : nakładka pozwalająca na śledzenie krok po kroku wykonywania się programu z edytora. W lewym górnym rogu mamy pomniejszony ekran, po jego prawej stronie panel sterujący, niżej okno z listingiem, i w końcu na samym dole widzimy okno ze zmiennymi i informacją o następnej instrukcji. Dzięki panelowi sterującemu, możemy wybrać jedną z czterech prędkości "odtwarzania" programu.

Pierwsza prędkość umożliwia analizowanie krok po kroku efektów wykomywania instrukcji, co widać na pomniejszonym ekraniku. Czwarta zaś prędkość - to już normalne oglądanie programu, tak jak by się go normalnie uruchomiło z edytora. To chyba najefektowniejszy i przy okazji najefektywniejszy debugger (choć nazwa wskazuje raczej na aspekt nauki, a nie poprawiania błędów) z jakim miałem się przyjemność dotychczas zetknąć.

Och, gdyby tak Asm-One mógł mieć coś tak działającego... Na otarcie łez użytkownikom starego Amosa powiem, że ich program również posiada coś w tym stylu, (choć jest to nie tak dopracowane). Nie wszyscy wiedzą bowiem, że w listingu programu możemy umieścić komendę Follow Zmienne (możliwe jest ich opcjonalne pominięcie) i od tego momentu możemy śledzić wykonywanie się programu i zmianę w jego trakcie wartości podanych przez nas zmiennych. Tryb śledzenia wyłącza się rozkazem Follow Off.

Help : Tu sprawa jest prosta i znana. Przykładowo zapomnieliśmy składni rozkazu Screen Open. Przesuwamy kursor na pierwszą literę komendy, uruchamiamy "Helpa" (z obszaru funkcji lub klawiszem Help) i z otwartego okienka z opisem instrukcji i jej składni, wyczytujemy wszystko co nas interesuje.

Mało tego : jeżeli nawet zapomnimy nazwy rozkazu, to Help nam pomoże! Piszemy pierwsze litery rozkazu np. Scr, uruchamiamy helpa i Easy Amos postara się domyśleć się o co nam chodzi. Oczywiście komend rozpoczynających się słowem Screen jest wiele, jednakże po tym jak sobie przypomnimy pierwszy wyraz, to zaczniemy eksperymentować z drugim np. Screen Op.

Do funkcji uzyskiwanych z klawiszem Shift włączone zostały (kosztem trzech innych) Enviroment, Bob Editor i Easy Disc. Po każdym naciśnięciu na Enviroment, zmieniają się kolory edytora. Każdy więc może wybrać coś dla siebie. Ja jednak po kilku pętlach przy ich przełączaniu, wróciłem do konfiguracji standardowej.

Wybranie Bob Editor'a powoduje uruchomienie programu do edycji bobów, a wybranie Easy Disc'a -programu w stylu Disk Mastera, czyli do przeglądania katalogów, kopiowania, kasowania i tym podobnych operacji na plikach. Easy Disc to bardzo wygodna pomoc przy pracy z Amosem, jednakże według mnie nie powinno tracić się cennego miejsca w obszarze wyboru funkcji, jeżeli można po prostu wgrać pewien program i uruchomić go.

Podobna uwaga tyczy się również Bob Editora. Najsmutniejsze jest to, że ofiarą padły takie funkcje jak Load Others czy New Others (nie występujące w ogóle w Easy Amosie), które to dotychczas służyły do obsługi programów w pamięci komputera.

Natomiast na plus Easy Amosa przemawia nowy, zdecydowanie lepszy file selector, czyli "wybieraczka" do plików. W starym Amosie jej poprzedniczka była utrapieniem przy pracy ze stacją dysków. Ponadto zmieniony został kształt strzałki na jakiś cywilizowany wygląd.

Dzieło wieńczą efektowniejsze requestry, wkomponowane graficznie do wyglądu edytora. Mimo niewątpliwych ulepszeń, nie można oprzeć się wrażeniu, że autorzy nie słyszeli o wszechobecnym już standardzie Newlook, uparcie lansując swoje własne rozwiązania. Lepszy jednak rydz niż nic.

Wydawać by się mogło, że w takiej sytuacji jedynym sensownym wyjściem jest oddanie na złom starego Amosa i używanie nowego. Nic bardziej błędnego. Jak uczy nas życie nie ma róży bez kolców, ale do rzeczy... Okazuje się, że Easy Amos bazuje na interpreterze znacznie okrojonym w stosunku do pierwowzoru. W praktyce oznacza to niemożność stosowania wielu instrukcji, których wcześniej się nauczyliśmy i z powodzeniem używaliśmy w programach.

Aby nie być gołosłownym, podaję kilka przykładów: nie działają między innymi instrukcje odnoszące się do coppera, dual playfieldu, skrolowania, Amala... Co najśmieszniejsze, edytor rozpoznaje te rozkazy nie sygnalizując błędu, a dopiero przy próbie uruchomienia pojawia się komunikat, że instrukcja nie jest dostępna w Easy Amosie.

Są jeszcze inne niespodzianki : z jednej strony wielu instrukcji zabrakło, a z drugiej strony dołożono nowe np. Track Load, czyli ładowanie do pamięci modułów SoundTrackera (chociaż nazwa sugeruje odczyt ścieżek) lub rozszerzono możliwości starych np. bezproblemowe otwieranie ekranów w trybie interlace. Tu zabierają, tam dają i gdzież tu sens?

Teraz trochę o dysku numer 2, czyli o przykładowych programach. Nie będę ich opisywać, wspomnę tylko, że wśród źródłówek można znaleźć ciekawe procedury na scrolling, efektowne wstawki przy pisaniu gier, oraz gotowe programy takie jak prosta baza danych i gry (Block Buster - odmiana Arkanoidu, Tricycle Race -wyścigi rowerami, Tricycle Ball - "piłka rowerkowa").

Zdecydowanie najciekawszy jeżeli chodzi o zawartość jest dysk trzeci Easy_Tutorial. Zawiera on programy takie jak seria TB_Step_.Amos (pokazująca sposób w jaki została napisana gra Tricycle Ball z dysku drugiego), Typing Tutor, oraz Challenge. Typing Tutor to nauka szybkiego odnajdywania liter na klawiaturze i nabierania wprawy w pisaniu.

Program Challenge to wspaniały test znajomości Easy Amosa sprawdzający nasze postępy w dwudziestu dziedzinach takich jak Tekst, Boby, Grafika, Klawiatura itp. Ciekawa oprawa graficzna i interesująco ujęte pytania testowe zapewniają nie tylko wymierne efekty w nauczaniu i sprawdzaniu wiedzy, ale i doskonałą zabawę. Aby uzyskać zaliczenie z jednego przedmiotu należy poprawnie odpowiedzieć na co najmniej 8 pytań (na jedną dziedzinę przypada ich 10).

Gdy indeks jest już do końca wypełniony "haczykami" zdanych egzaminów, następuje uroczyste wręczenie dyplomu Uniwersytetu Amosa. Dla leniwych lub mniej znających Easy Amosa podaję sposób na oszukanie profesorów. W procedurze CHECKGRAD znajduje się instrukcja :

```
If GRAD=1 Then Run PATH$+"Graduate.AMOS"
```

zamieniamy ją po prostu na :

```
Run PATH$+"Graduate.AMOS"
```

by po zakończonym jednym egzaminie (bez względu na rezultat) dostąpić zaszczytu ukończenia szkoły. Istnieje jeszcze metoda na zaniżenie poziomu studiów. W tej samej procedurze w instrukcji :

```
If GRADUATE(TEMP)<8 Then GRAD=0
```

zamiast ósemki podajemy na przykład trójkę. Od tej pory wystarczy zaliczyć wszystkie przedmioty z wynikiem przynajmniej 30% aby uzyskać tytuł amosowskiego magistra. W ten sam sposób można poziom nawet zawyżyć : wstawiając choćby 9. Uzyskanie 100% wyniku to spora sztuka, tym bardziej, że nie wszystkie pytania tyczą się ściśle Amosa.

Często musimy odpowiadać na pytania typu "Kto powiedział następujące słowa: (...)" i do wyboru mamy pośród innych, takie postacie jak Konfucjusz, Ronald Reagan czy Lenin.

To że w okrężny sposób otrzymacie dyplom wcale nie znaczy, że poradzicie sobie w pisaniu programów...

Pora na podsumowanie. Mimo wielu niedociągnięć Easy Amos to

naprawdę doskonały pakiet dla uczących się tego języka. Nasuwa się pytanie dlaczego autorzy celowo zubożyli interpreter skoro tak poprawiona wersja stałaby się z pewnością hitem w amosowskim światku.

Wydaje się być tylko jedna sensowna odpowiedź: otóż w przypadku stworzenia następcy Amosa, nikt by się już nie interesował starszą jego wersją. A w przypadku istniejącego stanu rzeczy, nabywca Easy Amosa wcześniej czy później będzie musiał przesiąść na ogólny standard, co wiąże się z koniecznością dokonania odpowiedniego zakupu, a więc większym zyskiem dla autorów.

Tymczasem cieszmy się i radujmy, gdyż mnogość przykładowych programów, bogata dokumentacja i staranne komentarze ułatwią poznanie zasad programowania nawet najbardziej odpornym na wiedzę Amigowcom.

Główną rolę odgrywa tu już samo nastawienie się twórców tej wersji programu głównie na naukę i przygotowanie adeptów do wkroczenia w świat "dorosłego" Amosa.

Zbigniew Piotrowicz.

O drukarkach kolorowych
słów kilka



Wielu z Was ma już drukarkę kolorową a wielu przymierza się do jej kupna. Tym którzy już mają drukarkę chciałbym przybliżyć sposób jej działania i zaprezentować inne modele, a tym którzy jej jeszcze nie mają pomóc w ewentualnym wyborze.

JAK DRUKUJE SIĘ W KOLORZE?

Biały kolor jest często stosowany jako symbol czystości. W rzeczywistości biały jest bardzo daleki od tej przysłowiowej czystości. Zmieszanie trzech podstawowych kolorów na jakie jest ludzkie oko czułe: czerwonego, zielonego i niebieskiego prowadzi dopiero do powstania koloru białego. Można to bardzo łatwo prześledzić np. przy ustawianiu preferencji na Workbench'u. Wszystkie inne kolory powstają dzięki odfiltrowaniu jakiejś części podstawowej barwy od białego światła.

Tak jak w telewizji drukarki aby uzyskać pełną gamę barw miksują trzy podstawowe kolory. Są nimi żółty, błękitny (blado-niebieski) i karmazynowy (odcień różowego). Czemu nie są to barwy używane w TV (czerwony, zielony, niebieski)? Jest to bardzo łatwo wytłumaczyć.

Tusz używany w drukarkach jest przeźroczysty i zachowuje się jak kolorowy filtr. Światło nie odbija się od niego lecz przechodzi przezeń i dopiero papier powoduje refleks. Odbite światło jest już odfiltrowane. Każdy z trzech tuszów jest zdolny do zatrzymania jednego z kolorów przepuszczając tym samym pozostałe dwa. Np. tusz żółty zatrzyma kolor niebieski lecz pozwoli przejść czerwonemu i zielonemu. Spróbujcie np. na Workbench'u lub na dowolnym programie graficznym ustawić suwaki kolorów na R (czerwony) = max, G (zielony) =max i B (niebieski) = min. Prawda, że powstał kolor żółty?!

Jeśli wydrukujemy dwie plamki tuszu na sobie to zostaną zatrzymane dwa kolory a jeden zostanie przepuszczony. I tak błękitny naniesiony na karmazynowy da nam w wyniku kolor niebieski. Odpowiednio błękitny i żółty da nam kolor zielony a karmazynowy i żółty da czerwony. Aby otrzymać kolor czarny należy zatrzymać wszystkie składniki światła czyli wydrukować wszystkie trzy barwy na sobie. Użycie więc kolorów błękitnego, żółtego i karmazynowego daje możliwość otrzymania pełnego zakresu barw, co byłoby niemożliwe przy kolorach czerwonym, zielonym i niebieskim.

Metoda uzyskiwania kolorów w drukarkach nazywana jest metodą "ujemnego" mieszania. W telewizji używa sie metody "dodatniego" mieszania. Zauważcie, że jeśli na swoim monitorze zmieszacie kolor niebieski, zielony i czerwony to otrzymacie biel. W drukarce odpowiednio błękitny, żółty i karmazynowy da czerń.

LICZĄC KOLORY

Jeśli uważnie przeczytaliście poprzednie słowa to z pewnością zauważyliście, że drukarka tak

naprawdę może wydrukować tylko siedem czystych kolorów: czerwony, zielony, niebieski, błękitny, karmazynowy, żółty i czarny. Niemożliwa jest niestety zmiana jasności tuszu. Nie można dodać więcej lub mniej koloru np. żółtego. Jego ilość jest zawsze taka sama i zależy od taśmy barwiącej.

Jedyną metodą osiągnięcia większej liczby kolorów jest tzw. DITHERING. Polega to na tym, że różnokolorowe punkty są umieszczane bardzo blisko siebie, jeden przy drugim. W efekcie daje to złudzenie nowego koloru. Za pomocą tej metody można wytworzyć bardzo wiele kolorów jednakże powstały w ten sposób "punkt" jest bardzo duży (oczywiście w porównianiu do normalnego punktu) i praktycznie drukowanie małych detali jest niemożliwe. Aby na szybko zobaczyć dithering poszukajcie w Kebabie dowolnego czarno-białego zdjęcia.

Zobaczycie kilka odcieni szarości a przecież farba ma tylko jeden kolor - czarny. Dithering w kolorze jest trudniejszy. Jest kilka metod grupowania punktów. Spróbujcie pod lupą obejrzeć kolorową okładkę Kebaba. Najprostszy dithering możecie zauważyć na reklamie Silver Dream'a na ostatniej stronie okładki Kebaba. Przyglądnijcie się dokładnie kolorowi ramki i napisów. W rzeczywistości użyta farba ma kolor jasno niebieski. Jak więc osiągnięto wrażenie ciemno-niebieskiego? Wystarczyło nałożyć na jasnoniebieską farbę siateczkę koloru czerwonego. Prawda, że proste?

Wielu producentów w reklamach podaje astronomiczne liczby kolorów, które ich drukarka może jakoby drukować. Często jest to nieporozumieniem. Producenci "zapominają" o dodaniu, że kolory te są uzyskiwane za pomocą ditheringu. Jednakże kilka typów drukarek może sterować jaskrawością podstawowych barw pozwalając na uzyskanie dużej ilości kolorów bez konieczności stosowania ditheringu. Np. drukarka "plujka" *Integer Colourcel* używa trzech różnych odcieni błękitnego,żółtego i karmazynowego. W wyniku mamy do dyspozycji ponad 100 kolorów. *EPL 8543* firmy *Panasonic* uzyskuje ich ponad 4000. Jedyną techniką jaka pozwala osiągnąć ogromną liczbę kolorów bez ditheringu jest metoda drukowania sublimacyjnego. Myślę, że liczba 16 milionów barw może zadowolić każdego, zwłaszcza, że oko ludzkie i tak odróżnia tylko około kilku set tysięcy barw.

Niektóre drukarki mają dodatkową czarną taśmę aby bez problemów uzyskiwać czarny kolor (w praktyce bowiem nałożenie trzech podstawowych barw nie daje idealnej czerni). Drukarki te są często reklamowane jako 15 kolorowe (bo jest możliwe 15 permutacji czterech kolorów). Jest to kolejne drobne oszustwo ze strony producenta bowiem z tych 15 kolorów aż 9 to różne odcienie szarości!! Inni producenci liczą sobie białą kartkę za kolor (przecież można nic nie wydrukować!) i podają cyfrę - 16 kolorów. Proszę więc uważać przy zakupie drukarki i nie dać się nabrać na nieprawdziwe reklamy.

ROZDZIELCZOŚĆ, NASYCENIE KOLORÓW I DOT-PATTERN

Oprócz ilości kolorów występują trzy inne ważne czynniki określające jakość wydruku. Pierwszym z nich jest rozdzielczość. Ogólnie rzecz biorąc im większa rozdzielczość tym lepiej. Jeżeli w instrukcji drukarki odczytamy, że rozdzielczość wynosi 150 dpi, to znaczy, że nasza drukarka potrafi wydrukować tylko 150 punktów na cal (dpi = Dots Per Inch). Wartość 600 dpi odpowiednio mówi, że nasza drukarka może na jednym calu umieścić, aż 600 punktów. Nawiasem mówiąc będzie to pewnie drukarka laserowa wysokiej klasy!

Nasycenie kolorów jest trudne do zmierzenia. Jeśli patrzymy na wydruk i widzimy mocne, jaskrawe kolory to znaczy, że nasycenie jest duże. Kolory słabe i wyblakłe oznaczają nasycenie małe. Oczywistą sprawą jest, że lepiej mieć kolory o dużym nasyceniu bo w razie potrzeby można je stonować i otrzymać te bardziej wyblakłe. Wyblakłych kolorów już nie da się bardziej nasycić.

Trzecim czynnikiem jest tzw. "dot-pattern", czyli odpowiednie ułożenie punktów na wydruku. Jak już powiedziałem kolory wytwarza się za pomocą ditheringu czyli odpowiedniego układania punktów tak aby stworzyć nowy kolor. To właśnie ułożenie ma zaskakujący wpływ na jakość wydruku. Metod ditheringu jest bardzo wiele i jest to w zasadzie temat na odrębny artykuł. Postaramy się tym zająć w przyszłości. Tymczasem znowu odsyłam Was do zdjęć w gazetach - można tam dithering zaobserwować w całej krasie!

Wreszcie dotarliśmy do samych drukarek. W tej chwili na rynku jest 5 typów drukarek kolorowych. Nie, Nie! Proszę tego nie mylić z modelami! Samych modeli jest setki, lecz każdy z nich korzysta z jednej z pięciu dostępnych metod uzyskiwania kolorowego druku.

DRUKARKI MATRYCOWE

Jest to metoda najtańsza. Prawie trzecia część drukarek matrycowych posiada możliwość drukowania w kolorze. W niektórych modelach jest to tylko sprawą zamocowania kolorowej taśmy. W innych koszt przeróbki waha się w granicach od 900 tys. do 2,5 miliona zł. Kolorowe drukarki matrycowe używają szerokich taśm z barwnymi pasami biegnącymi wzdłuż nich. Jedna linia wydruku jest uzyskiwana w czterech przejściach - po jednym dla każdego koloru. Pomiędzy każdym kolejnym przejściem specjalny mechanizm podnosi taśmę barwiącą tak aby kolejny kolor znalazł się pomiędzy głowicą a papierem.

Drukarki matrycowe są tanie ale też i wydruki z nich uzyskiwane nie są najlepsze. Tusz umieszczany w taśmach barwiących nie jest w stanie dać dobrych, mocnych kolorów. Nawet czarna taśma nigdy nie da idealnej czerni. Zawsze będzie to ciemniejszy lub jaśniejszy jej odcień. Innym problemem są wyraźnie zauważalne poziome linie biegnące wzdłuż litych płaszczyzn koloru. Często drukarki pozostawiają białe linie pomiędzy każdym przejściem głowicy. Same taśmy barwiące również sprawiają dużo kłopotu. Jeśli np. wydrukujemy ekran z przewa-

gą koloru żółtego to może się okazać, że tak bardzo zużyliśmy go, że cała taśma jest do wyrzucenia, mimo iż pozostałe kolory są jak najbardziej w porządku! Inną wadą jest strasznie głośna praca. Drukowanie czegoś w nocy potrafi obudzić sąsiadów!! (sam to przeżyłem!)

Z tych powodów drukarek matrycowych nie polecam osobom które będą dużo drukowały lub potrzebowały dobrej jakości. Drukarki te są najlepsze do okazjonalnej pracy i dla ludzi bez wielkich wymagań.

Jak i przy normalnych drukarkach i tu mamy wybór między modelami 9 i 24-igłowymi. Drukarki 24 igłowe dają lepszą rozdzielczość a tym samym o wiele lepszą jakość druku. Niektóre modele osiągają teoretyczną rozdzielczość 360 dpi. Ktoś może powiedzieć, że to więcej niż drukarki laserowe. W rzeczywistości jednak wydruk z każdej "laserówki" jest wyraźnie lepszy od wydruku z najlepszej drukarki igłowej. Mimo tego modele 24 igłowe na pewno są lepsze od 9 igłowych, które dają tylko rozdzielczość 180 dpi.

W czasie eksploatacji innym ważnym kosztem jest cena jednej zadrukowanej kartki. Trzeba przyznać, że drukarki matrycowe są najtańsze. Jedna strona kosztuje bowiem od około 100 zł do 1000 zł.

Zalety: tania, drukuje na niemal wszystkim

Wady: słabe kolory, wolna, głośna, słaba rozdzielczość

DRUKARKI "PLUJKI"

Drukarki tego typu jak sama nazwa wskazuje "wypluwają" pod dużym ciśnieniem krople tuszu na papier. Jedną z technik takiej pracy jest "bubblejet" stosowana przez firmę *Canon*. Polega ona na gwałtownym podgrzaniu drobinki tuszu powodując jego odparowanie i wypryśnięcie przez specjalną dyszę.

Jedną z zalet tych drukarek jest to, że głowica drukująca może być robiona techniką podobną jak przy produkcji układów scalonych. Dzięki temu głowice z dużą ilością dysz są bardzo tanie. Niektóre z modeli mają wbudowane głowice drukujące do zbiorników z tuszem!

Plujki używają oddzielnego zestawu dysz do każdego koloru. Oznacza to, że wszystkie trzy lub cztery kolory mogą być drukowane za jednym przejściem. Zwiększa to znacznie szybkość pracy. Wielką zaletą tych drukarek nad drukarkami matrycowymi jest stała dobra jakość koloru. Nie występuje tu proces blaknięcia tuszu w miarę jego zużywania. Niestety i tu czasami występują poziome linie wzdłuż wydruku.

Drukarki te mają jednak i swoje wady. Duże płaszczyzny jednolitego koloru stają się po prostu mokre, co może spowodować zmarszczenie się papieru i w efekcie roztarcie tuszu przez głowicę drukującą. Inną wadą jest papier. Drukarki te owszem drukują na normalnym papierze, jednak najlepsze rezultaty osiąga się stosując jego specjalny rodzaj.

Jakość druku uzyskiwana przez różne modele jest bardzo różna. Wiele pracuje na niskiej rozdzielczości około 180 dpi. Inne uzyskują już od 300 do 360 dpi. Niektóre dają bardzo dobry kolor a słaby text, inne odwrotnie. Chyba najlepszą drukarką czarnobiałą jest *Hewlett Packard Deskjet 500.* HP Painjet daje z kolei bardzo dobre kolory lecz przy niskiej rozdzielczości.

Nowa wersja *HP Deskjet 500C* pozwala już na osiągnięcie 300 dpi. Bardzo dobrą drukarką jest *Integrex Colourjet* (poprawiony model Canon'a PJ1080A), który ma osiem dysz : czarną, żółtą, trzy na różne odcienie błękitnego i trzy na różne odcienie karmazynowego. Daje to ponad 100 kolorów bez dithering'u lecz niestety przy rozdzielczości tylko 160 dpi.

Zalety: Niedrogie, dość dobre kolory

Wady: Smużenie kolorów, wolne, polecany specjalny papier, niezbyt wysoka rozdzielczość

Koszt strony: 1000 - 2000 zł.

DRUKARKI TERMICZNE

Wiele osób myli drukarki termiczne z laserowymi. Wyglądają one dość podobnie jednak pracują na zupełnie innych zasadach. Drukarki termiczne używają rolek plastikowej folii pokrytej warstwą kolorowego wosku. Głowica drukująca pokryta tysiącami elementów grzejnych porusza się wzdłuż papieru topiąc nań wosk. Kiedy jeden kolor jest już położony proces powtarza się dla innego. Kolory uzyskiwane tą metodą są bardzo dobrze nasycone a dobrej jakości głowica pozwala na osiągnięcie rozdzielczości 300 dpi.

Nadrukowanie na sobie trzech podstawowych kolorów daje bardzo dobrą czerń. Często więc aby zmniejszyć koszty stosuje się trzykolorowe taśmy. Efektem ubocznym są wtedy frędzelki na brzegach małych detali. Dla wydruków o bardzo dużej jakości stosuje się więc taśmy czterokolorowe.

Wydrukowanie jednej strony A4 zabiera około jednej minuty. Drukowanie z taśmą czterokolorową czas ten przedłuża.

Jedną z większych wad jest brak możliwości drukowania na skrajach stron. Wynika to z tego, że drukarka musi po prostu trzymać za coś papier. Metodą na usunięcie tej wady jest stosowanie papieru o kilka centymetrów większego od normalnego.

Drukarki termiczne za bardzo nie nadają się do zastosowań biurowych. Cena za stronę jest dość duża, praca wolna a ich lśniący druk może wyglądać dziwnie na np. liście płac. Wszystkie drukarki tego typu pracują z rozdzielczością 300 dpi. Jedynym wyjątkiem jest chyba *Panasonic EPL 8543* który ma tylko 203 dpi.

Zalety: Bardzo dobre kolory, dobra rozdzielczość

Wady: Duża cena, dość wolne, konieczny specjalny papier, duże obszary nie dające się zadrukować.

Koszt strony: 7000 - 15000 zł.

DRUKARKI SUBLIMACYJNE

Wielu ludzi potrzebuje wręcz fotograficznej jakości wydruków. Oznacza to, że będą musieli zakupić drukarkę sublimacyjną. Często jednak po stwierdzeniu jak wielkie są koszta zakupu i eksploatacji takich drukarek okazuje się, że doskonała jakość druku wcale nie była taka potrzebna.

Drukarki sublimacyjne mają wiele wspólnego z drukarkami termicznymi. Oba typy używają rolek taśmy z barwnym woskiem i tysięcy elementów grzejnych w głowicy drukującej. Różnią się jednak tym, że zamiast topić wosk drukarki sublimacyjne powodują jego odparownie i kondensacje na papierze. Stąd też wzięła się nazwa - proces zmiany ciała stałego w lotne nazywa się sublimacją.

Największą zaletą tej metody jest to, że drukarka może sterować jaskrawością koloru przenoszonego na papier. Każdy z kolorów może mieć aż 256 odcieni. Stąd też kombinacja trzech podstawowych kolorów daje ponad 16 milionów barw możliwych do uzyskania bez dithering'u.

Technika sublimacyjna jest bardzo droga. Wielu producentów decyduje się więc wypuszczać modele które mogą drukować na małym papierze (np.A6). Rozdzielczość tych drukarek wynosi tylko około 155 dpi. Może wydawać się to mało, jednak ogromna ilość kolorów bez dithering'u robi swoje. Skanowane obrazy po wydruku są niemal idealnie wierne oryginałom. Ekrany przeniesione z kamery video są po prostu ich kolorowymi fotografiami. Niska rozdzielczość daje się dopiero we znaki przy drukowaniu małych detali. Istnieją również drukarki osiągające 300 dpi. Kosztują one jednak prawie dwa razy tyle co ich odpowiedniki o niskiej rozdzielczości. Drukarki sublimacyjne są wolne. Jedna strona zabiera ponad pięć minut! Największym problemem podczas samej pracy jest rozmiar zbioru z danymi do jednej strony. Strona A4 przy rozdzielczośći 150 dpi zajmuje około 6Mb pamięci! Dodanie czwartego podstawowego koloru (czarnego) i podniesienie rozdzielczości do 300 dpi zwiększa rozmiar zbioru do 32 Mb.

Zalety: Wręcz wspaniała jakość kolorów bez dithering'u, dobra rozdzielczość

Wady: Duża cena, bardzo duża cena za stronę, bardzo wolne, konieczny specjalny papier

Koszt strony: 40.000 (czterdzieści tysięcy!!!) zł.

DRUKARKI LASEROWE

Tak naprawdę to kolorowe drukarki laserowe nie istnieją! To co ludzie nazywają "laserówką" jest po prostu kolorową laserową kopiarką z komputerowym interface'm. Niektóre kolorowe kopiarki używają tradycyjnego optycznego systemu kopiowania i jako takie nie nadają się na drukarkę. Inne modele używają cyfrowego skanera do przetwarzania obrazu który może być natychmiast wydrukowany. Dodanie specjalnego interface'u pozwala na przesłanie komputerowego ekranu na kopiarkę.

Pionierem w tej technice jest *Canon* ze swoją *CLC 300* i *500* (Colour Laser Copier). Obie maszyny są formatu A3 i pracują z rozdzielczością 400 dpi. Każdy z podstawowych kolorów może mieć aż 256 odcieni! Teoretycznie drukarki te mogą więc dorównać drukarkom sublimacyjnym. W praktyce jednak jakość nie jest tak dobra. Laserowy kopier bardziej zmienia wielkość punktu niż jego odcień.

Ekran formatu A3 o rozdzielczości 400 dpi zajmuje około 128 Mb pamięci. Tak duża konieczna pamięć jeszcze bardziej powiększa i tak ogromną cenę urządzenia.

Zalety: Wysoka jakość, duża szybkość pracy, duża rozdzielczość, niska cena za stronę, dobre kolory

Wady: Ogromna cena, duże rozmiary, duża energochłonność

Koszt strony: 1000 zł.

Po przeczytaniu: Computer Buyer, napisał

Sambor Kuźma

FORUM



Ostatnio coraz więcej czytelników pyta w listach czy nie można otworzyć nowej rubryki, która przynajmniej po części rozwiąże problem "jak przejść grę"?

Ponieważ sami nie jesteśmy w stanie spełnić wszystkich oczekiwań Czytelników, stąd nasza nowa rubryka: **FORUM**

Jeżeli posiadacie jakieś "chwyty do gier",napisaliście jakiś programik w BASIC'u lub w asemblerze i uważacie, że innym mógłby się on przydać lub spodobać napiszcie na adres redakcji.

Mamy nadzieję, że zainteresowanie tą rubryką będzie wystarczająco duże i czekamy na listy, które muszą być podpisane imieniem, nazwiskiem i adresem gdyż tylko takie będą mogły znaleźć się w naszym FORUM.

Na początek chciałbym przedstawić Czytelnikom mój własny zestaw POKI & PEEKI:

`POKE 53280,peek(53281)` - ujednolica tablicę kolorów

POKE 646,x - zmiana koloru kursora
POKE 53265,80 - kursor jest czerwony,a zapis pozostaje jasnoniebieski
POKE 56325,0- zwalnia pracę komputera
POKE 56325,x- zmienia czestotliwość migania kursora
POKE 56325,51- normalna praca komputera

Sposoby na ciekawskich:

POKE 775,200-blokuje rozkaz LIST
POKE 775,167-odblokowuje ten rozkaz
POKE 774,0-zostają wyświetlone tylko numery lini programu
POKE 774,26-wszystko chodzi normalnie
POKE 774,27-groch z kapustą
POKE 22,35-wyświetla program bez numeru lini
POKE 22,25-wszystko w normie
POKE 120,2-nie przyjmuje żadnych rozkazów
POKE 770,100-spróbuj teraz go zatrzymać,po komendzie LIST
POKE 768,143-po skończeniu programu wykonuje RESET
POKE 53265,11-obraz znika,ale jedo zawrtość pozostaje w pamięci
POKE 53265,27-powoduje pojawienie sie obrazu
POKE 774,226:POKE 775,252-sekwencja ta po rozkazie LIST niszczy program

Zabawa z klawiaturą:

POKE 649,0 lub POKE 655,71-blokuje całą klawiaturę
POKE 649,10 lub POKE 655,72-odblokowuje klawiaturę
POKE 808,239-RUN/STOP nie reaguje
POKE 808,255-RUN/STOP i RESTORE nie działają
POKE 808,237-wszystko działa normalnie
POKE 788,52-blokuje RUN/STOP
POKE 788,49-odblokowuje RUN/STOP
POKE 792,193-blokuje RESTORE
POKE 792,71-odblokowuje RESTORE
POKE 792,226:POKE 793,252-po wciśnięciu RESTORE powoduje RESET
POKE 788,226:POKE 789,252-to samo tylko, po wciśnięciu klawisza RETURN

Ochrona przed PIRATAMI:

POKE 818,252:POKE 819,253:
POKE 808,225 lub
POKE 818,34:POKE 819,252:
POKE 808,225

sekwencja ta uniemożliwia nagranie programu na dysku lub kasecie.

Paweł Piętka

"Ancient Battles" - Starożytne Bitwy



Ancient Battles - doskonały produkt pana R.T.Smith'a. Gra została stworzona w latach 1988-1990. Jest to typowa gra strategiczna o bardzo rozbudowanej fabule, całkiem niezłej grafice i żadnej muzyce. Gracz wciela się w postać jednego z wielkich dowódców starożytności, może być Darius'em wodzem Persjan, Aleksandrem Macedońskim, czy też Asterixem Galem.

Program został wykonany z wielką starannością pod względem logicznym i przebija z niego wielka wiedza autora o taktyce i "militariografii" starożytnej. Możliwa jest rozgrywka pomiędzy dwoma graczami, komputerem i graczem lub komputerem i komputerem. Grać można za pomocą myszy lub klawiatury korzystając z kursorów i klawiszy *enter* oraz *escape*.

Gdy już po tej krótkiej zapowiedzi zdecydujemy się zasiąść do Starożytnych Bitew pierwszą rzeczą na jaką należy zwrócić uwagę jest główne menu. Dostępnych mamy w nim pięć opcji. **Begin New Game** jest to po prostu rozpoczęcie nowej rozgrywki, z możliwością wyboru nowych armii i mapy jako terenu przyszłej bitwy. **Play Scenario** (gra według scenariusza), jest to gra według wcześniej ułożonego i zapisanego na dysku scenariusza. **Play Saved Game** (kontynuacja wcześniej rozpoczętej gry), ten podpunkt nie wymaga chyba wyjaśnień. **Design Map** (układanie mapy), bardzo ciekawa opcja umożliwiająca graczowi na samodzielne ułożenie terenu bitwy.

Design Map - po uruchomieniu tej opcji pojawia się nam nowe menu. Możemy w nim ujrzeć nowe opcje: **Use Curent Map** (używanie aktualnej mapy), która uaktywnia mapę przez nas zrobioną lub wgraną, powodując, że na niej właśnie odbędzie się najbliższa bitwa. **New Map** (tworzenie nowej mapy), dzięki temu możemy stworzyć zupełnie nową mapę. Tym różni się to od Edit Map, że podajemy wymiary mapy (jej szerokość i długość), a następnie przechodzimy do edycji form terenu.

Edit Map (edycja Mapy) umożliwia nam zmianę ukształtowania terenu gotowej już mapy. Z dostępnych nam form terenu takich jak: różne rodzaje kompleksów leśnych, wzgórza, rzeki, przeszkody (kamienie, bagna, zarośla) i skaliste wzgó-

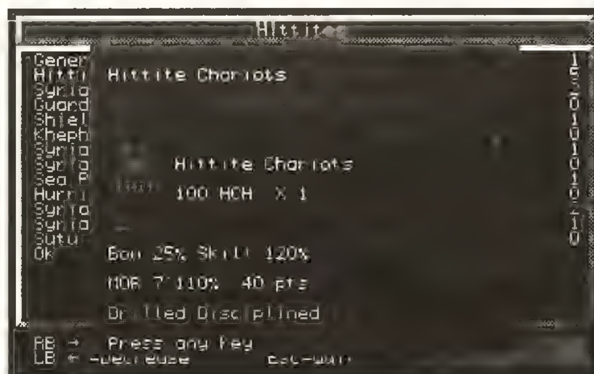


rza możemy stworzyć bardzo interesującą scenerię dla naszej rozgrywki batalistycznej.

Natomiast jeżeli któryś z tych elementów nam nie odpowiada, to możemy usunąć go za pomocą funkcji **Remove terrain** (usunięcie terenu). Każdy z obiektów na mapie ma swoje znaczenie. Wiadomo, że w lesie lepiej będzie się sprawowała lekka piechota od np. ciężkiej kawalerii. Natomiast konnica zyska przewagę atakując ze zbocza wzgórza.

Należy też wziąć pod uwagę przy dyslokacjach jednostek, że np. rzeka czy bagno zwalnia przemarsz żołnierzy, a co się z tym wiąże przybędą oni później na miejsce przeznaczenia. Jest to bardzo ważne przy przygotowaniu dokładnego ataku. W tym menu możemy jeszcze nagrać wykonaną przez nas mapę (Save Map) lub wgrać inny projekt (Load Map). Jeśli chodzi o tworzenie mapy to sądzę, że ta niewielka ilość informacji wystarczy w zupełności gdyż nie jest to skomplikowane.

Po wybraniu opcji nowej gry, komputer pyta nas za ile punktów będziemy kupowali nasze jednostki (750 pts ,1000 pts, 1500 pts, 2000 pts, Unlimited pts). W zależności od naszego wyboru będziemy mogli zakupić mniej lub więcej oddziałów. Wpłynie to oczywiście na długość rozgrywki. Wypróbowanym przeze mnie ilością jest 1500 pts, gra się nie dłuży, a jednocześnie możemy przeprowadzić całkiem niezłą batalię. Wybranie Unlimited pts spowoduje, że przy wyborze naszych wojsk będziemy ograniczeni jedynie maksymalną ilością jednostek w danej armii.



Kolejną czynnością jaką musimy wykonać jest wybór armii. W pierwszej kolejności wybieramy armię gracza pierwszego a potem gracza drugiego. Do wyboru mamy kilkadziesiąt armii. Są to między innymi armie z całego przekroju historycznego państwa rzymskiego, czyli Galowie, Parthianie, Germanie, Hunowie, Wizygoci, Grecy i wiele innych. Oczywiście obaj gracze mogą wybrać te same nacje lecz będzie to bardzo mało interesująca rozgrywka.

Po wyborze armii przechodzimy do szczegółowego wyselekcjonowania jednostek. Ich wybór jest oczywiście uzależniony od armii, którą wybraliśmy. Logiczne, że w armii greckiej nie będzie legionów, w armii rzymskiej hypaspistów, a w armii macedońskiej oryginalnej falangi greckiej. Każda armia posiada swoje charakterystyczne jednostki. Wybór odpowiedniej nacji w dużym stopniu uzależnia możliwość zwycięstwa. Np. falangi greckie nie będą miały wielkiego kłopotu z rozbiciem Galów lecz będą już miały ciężki orzech do zgryzienia gdy spotkają się z legionami rzymskimi. Na temat poszczególnych zależności między armiami można było by bardzo wiele napisać jednakże najlepiej wypróbować sprawność każdej z nich w samodzielnej grze.

Po wybraniu składu armii kolejną czynnością jest ustawienie wojska przed bitwą. Do wyboru mamy dwie opcje:

Deploy Forces - to właśnie umożliwia nam rozlokowanie żołnierzy przed bitwą, **Change Settings** - tutaj możemy ustawić widoczność (Poor - słaba, Normal - normalna, Night -noc), zmienić ilość tur w ciągu których będziemy toczyć naszą batalię i ustawić sposób w jaki będą rozstawiane nasze jednostki.

Normal Deployment - rozkład jednostek pozostaje normalny tj. w zależności od wielkości mapy zajmowane są rzędy z obydwu końców, natomiast **Free Deployment** powoduje, że zajmować możemy wszystkie pola na mapie.

Po wybraniu opcji Deploy Forces komputer spyta nas o to czy chcemy samemu rozstawiać nasze wojska (Manual Deployment), czy też ma zrobić to za nas komputer (Auto Deployment). Układanie jednostek samodzielne polega na tym, że klikamy na oddział który chcemy ustawić, wybieramy rozkaz Deploy, a następnie najeżdżamy kwadracikiem na miejsce docelowe i jeszcze raz klikamy (lewy przycisk myszy lub enter).

Do pomocy mamy opcje Deploy Left i Deploy Right, które automatycznie ustawiają jednostkę na lewo lub prawo od ostatniego ulokowanego oddziału. Z trybu ustawiania wychodzimy klikając na escape lub wybierając opcję End Deployment. Jeżeli chcemy by nasze wojska zostały rozstawione przez komputer, może on to zrobić w zależności od armii na różne sposoby.

Komputer pokazuje nam optymalne przykłady rozstawienia naszych wojsk np. DEFEND-CHARGE-DEFEND czy DEFEND-ATTACK-SKIRMISH. Jeżeli ustawimy skrzydła na ATTACK to będzie się w nich znajdowała konnica, jeżeli ustawimy centrum na ATTACK to będą się w nim znajdowały jednostki słoni, cataphracty i rydwany. ADVANCE oznacza atak z użyciem piechoty, SKIRMISH z użyciem lekkiej konnicy i lekkiej piechoty, DEFEND pozostałe jednostki, najczęściej mieszane, przygotowane są do obrony, a nie do ataku.

Następnie ustalamy na jakiej mapie odbędzie się walka. Mapa przez nas wybrana to Current Map, i ustalamy kto będzie kierował jedną i drugą armią.

Tyle z samego przygotowania do rozgrywki. Abu móc poznać i zrozumieć wartość każdej jednostki musimy zorientować się w określających je danych. Część tych danych widzimy na głównym oknie informacyjnym oddziału, resztę możemy uzyskać po wybraniu opcji Report. Możemy więc dowiedzieć się o: - nazwie jednostki
- imieniu wodza pod którego komendą znajduje się dany oddział
- typie jednostki
- posiadanym rodzaju zbroi, ewentualnie posiadanej tarczy

Jednostka może nie posiadać żadnej zbroi, posiadać lekką lub ciężką, kawaleria może mięć również zbroje dla konia (Barded). Jeżeli na wyposażeniu oddziału znajduje się broń umożliwiająca strzelanie, to pokazane jest jak wiele osób ją posiada i jakie jest wyszkolenie żołnieży w posługiwaniu się nią (Skill). Rodzaje broni strzelającej to łuki (Bows), włócznie (Javelins, JAV), proce (Slings, SI) i typ mieszany, czyli łuki i włócznie (Mixed, MIX).

STR...Strenght - pokazuje siłe oddziału w ludziach.

FAT...Fatigue - podaje poziom

zmęczenia oddziału. Jest to związane z efektywnością i morale. Jednostka mocno zmęczona łatwo wpada w panikę i ucieka.

DIS...Discipline - jednostką zdyscyplinowaną jest łatwiej kierować. Jednostki Drilled reagują szybciej od innych. Oddziały dzikie (Wild) trudniej zmusić do ucieczki. W informacji dowiadujemy się również jak wydajny jest oddział w walce wręcz.

Typy jednostek: Generał (GEN)- nie walczy, lecz może być przyłączany do innych jednostek w celu podniesienia ich morale i wydajności walki.

Ciężka konnica (HC, Heavy Cavalry) - jednostka szybkiego reagowania. Stanowiła w wielu państwach trzon armii. Nie powinna być używana w trudnym terenie. Bardzo słaba w starciu z ciężką piechotą uzbrojoną w włócznie i piki.

Lancersi (HCL, Lancers) - ciężka konnica uzbrojona w lance. Używana do wspierania HC w walce z ciężką piechotą.

Lekka konnica (LC, Light Cavalry) - często używana w górach, wykorzystywana do spychania wroga w dół i dobijania.

Cataphract konnica (CAT) - mężczyźni w bardzo ciężkich zbrojach, jeżdżący w zamkniętych formacjach, bardzo silni w ataku lecz słabej zwrotności. Często wykorzystywani w armiach orientalnych.

Ciężka piechota (HI, Heavy Infantry) - oddziały piechoty walczące w zamknietej formacji, używane w trudnym terenie.

Oszczepnicy (SP, Spearmen) - jednostki piechoty wyposażone w długie oszczepy lub piki, po raz pierwszy wykorzystywani w greckich falangach, dobrzy w obronie, przeciw konnicy i wykorzystywani w dużej liczbie.

Barbarzyńska Piechota (BI, Barbarian Infantry) - najczęściej wykorzystywana w trudnych warunkach, łatwo atakują we wszystkich kierunkach, doskonałe do wsparcia ciężkiej piechoty.

Lekka piechota (LI, Light Infantry) - piesi wojownicy walczący w luźnych formacjach, generalnie słabi, wykorzystywani na trudnych terenach.

Słonie (EL, Elephants) - słonie tworzą bardzo silną formację ofensywną, nie są ruchliwe, łatwo ulegają panice, konnica nie może walczyć ze słoniami ponieważ konie boją się tych zwierząt.

Ciężkie rydwany (HCH, Heavy Chariots)- specjalne bojowe rydwany z liczną załogą, używane do szokowania armii przeciwnika, we wczesnych armiach generalnie wykorzystywane do wspierania działań ciężkiej konnicy, później wykorzystywane w trudnym terenie.

Lekkie rydwany (LCH, Light Chariots) - małe rydwany ze zredukowaną załogą.

Scytyjskie rydwany (SCH, Scythed Chariots) - ciężkie rydwany z przyczepionymi do kół włóczniami, wymyślone do przerywania formacji piechoty nieprzyjaciela, w momencie rozpoczęcia ataku w chwilę przed zderzeniem kierowcy wyskakiwali z pojazdów, w praktyce wykorzystywane do ataku na słonie, które panikując tratowały własne wojsko.

Każda jednostka w zależności od rodzaju w ciągu jednej tury wykonuje określoną liczbę kroków.

HI,SP,BI	dwa kroki
LI,EL,HCH,SCH,SHC	trzy kroki
HC,HCL,LCH	cztery kroki
LC,GEN	pięć kroków

W czasie walki często zdarza się, że jakiś oddział wpada np. w las, bagno czy na wzgórze. Ma to duże znaczenie na wiele czynników.

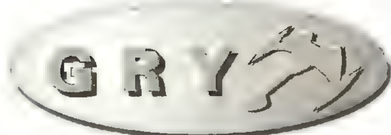
Pustynia (Desert) - przyspiesza walkę Lasy (Woods) - ukrywają jednostki znajdujące się za nimi, powoduje osłabienie walki, doskonale w lesie czują się lekka oraz barbarzyńska piechota. Wzgórza (Hills) - ukrywa jednostki znajdujące się za nimi, zwiększa zdolności bojowe wszystkich jednostek. Góry (Mountains) - podobne do wzgórz lecz zmniejszają zdolności bojowe jednostek oprócz lekkiej oraz barbarzyńskiej piechoty. Zarośla (Shrub) - podobnie jak las lecz ukrywają jedynie lekką piechotą Bagna (Swamp) - redukują efektywność jednostek poza lekką i barbarzyńską piechotą, uniemożliwiają przejście koniom. Rzeka (River) - redukuje zdolności bojowe wszystkich jednostek znajdujących się na niej.

Zwycięstwo? Trudno ustalić w końcu kto zwyciężył, zależy to od indywidualnej umowy graczy. Można walczyć, aż jeden z graczy straci 50% stanu osobowego armii, czy też gdy na mapie nie pozostanie żadna jednostka wroga, ale to już zależy od Was. Walkę można kontynuować dnia następnego, oczywiście nie dosłownie, po każdej turze zamiast enter można wcisnąć Esc, który przeniesie nas do menu. W nim możemy skończyć walkę czy też ją kontynuować, skorzystać z opcji Debriefing. Z niej otrzymamy trochę informacji statystycznych o tym ile żołnierzy zgineło, ile uciekło, a ile pozostało na polu bitwy by dalej walczyć o naszą cześć.

To wszystko, na szczęście, jest wpół do drugiej w nocy i ostatnie literki wklepuje językiem podtrzymując powieki, a czołem czule kleję się do obudowy mojej AMIGI.

Marcin "Curl" Kasprzak





GRY NA C-64

LETHAL WEAPON (c) OCEAN

Kolejna labiryntówka firmy OCEAN. I jak zwykle w przypadku tej firmy, jest to gra na podstawie filmu. Ogólny wygląd gry jest zbliżony do HUDSON HAWK'a, ale w LETHAL WEAPON mamy do czynienia ze znacznie lepszą grafiką i bardzo płynną animacją. Niestety gra ta nie zawiera zbyt dobrej muzyki, ale mnogość poziomów powinna to wszystkim zrekompensować.

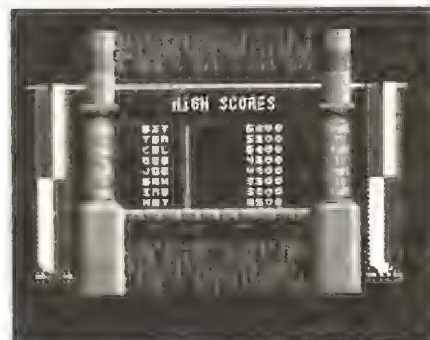
RAMPART (c) DOMARK

Jedna z ciekawszych gier, jakie ukazały się w ostatnim miesiącu.



Łączy ona w sobie elementy zręcznościowe i strategiczne. Naszym zadaniem w grze jest obrona fortecy przed atakami wrogów. Cały czas gry jest podzielony na tury, w których możemy naprawiać mury, ustawiać armaty, itd.

To są cechy wspólne z grami strategicznymi. Natomiast część zręcznościowa zaczyna się w momencie, gdy strzelamy do wrogów.



Ciekawy pomysł, słaba muzyka, grafika bez zbytnich "fajerwerków", ale czytelna. W sumie gra warta polecenia.

UGH

Konwersja dość popularnej gry z Amigi. Rzecz polega na przechodzeniu kolejnych plansz w których musimy transportować (gramy tu rolę czegoś podobnego do małego helikopterka) małe ludziki z półki na półkę. Gra jest dość słabo dopracowana, ale charakteryzuje się dość dużą "wciągalnością".

INDIANA JONES IV (c) LUCASFILM GAMES

Nie jest to jakby się można było spodziewać konwersja amigowskiej



gry przygodowej, lecz zupełne inna gra niż jej wspomniany odpowiednik. Jest to trójwymiarowa gra komnatowa (z tym, że komnaty nie są zdecydowanie od siebie oddzielone, lecz czasami kilka z nich stanowi jedno pomieszczenie. W grze tej sterujemy dwiema osobami tzn. Indym i jego dziewczyną a sens gry przypomina trochę Last Ninję. Gra jest jednak trochę niedopracowana (brak muzyki itp.)



WWF 2 - EUROPEAN RAMPAGE (c) OCEAN

Druga część oceanowskiego "symulatora" wrestling'u. Dlaczego słowo symulator umieściłem w cudzysłowie?



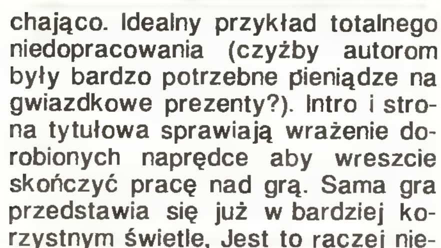
Zrobiłem to ze względu na niską jakość gry. Moim zdaniem pierwsza część była o wiele lepsza. Wadą tej gry jest mała ilość ciosów i bardzo małe figurki graczy. Grafika jest dość dobra, ale w tego typu



grze, to żadna rekompensata. Polecana tylko dla prawdziwych fanatyków "nawalanek".

DALEK ATTACK (DOCTOR WHO) (c) ADMIRAL SOFTWARE

Strona tytułowa tej gry działa na wszystkich zdecydowanie odpy-



chająco. Idealny przykład totalnego niedopracowania (czyżby autorom były bardzo potrzebne pieniądze na gwiazdkowe prezenty?). Intro i strona tytułowa sprawiają wrażenie dorobionych naprędce aby wreszcie skończyć pracę nad grą. Sama gra przedstawia się już w bardziej korzystnym świetle, Jest to raczej nietypowa labiryntówka, gdyż labirynt stanowią różne miasta. Może animacja nie jest zbyt płynna, ale gra nie jest zła. Ogólnie średni program za średnie pieniądze.

FIRST SAMURAI (c) IMAGE WORKS

Jedna z ładniej dopracowanych graficznie gier jakie ukazały się w tym miesiącu. Wcielamy się w skórę tytułowego "Samuraja" i mamy do przejścia cztery bardzo długie poziomy na których walczymy z wszystkim co się rusza (i nie tylko). Programiści ułatwili nam trochę zadanie w postaci kodów, które program podaje po zakończeniu każdego poziomu. Pozwala to na zaczynanie gry od poziomu na którym się skończyło, a nie od samego początku. Dużo dobrej zabawy. Zdecydowanie polecam wszystkim wielbicielom gier zręcznościowych.



NFL GOLF (c) GRANDSLAM

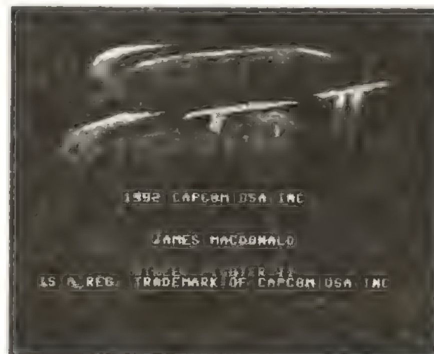
Kto wie, może to najlepsza symulacja gry w golfa, jaką napisano na C64? Oprócz tradycyjnego, wektorowego rysowania tła, mamy tu też doczynienia z dużą ilością obiektów, które zwiększają realność widoku jak np. drzewa. Do



wyboru mamy kilkanaście tras i wiele różnych kijków. Opanowanie obsługi programu może nastręczyć pewne trudności, ale już po krótkim treningu wszystko staje się jasne. Grafika stoi na wysokim poziomie brak jest natomiast muzyki.

THE ORMUS SAGA (c) DORGO GAMEPLAY

Kolejna z licznej serii role-playing-games (w skrócie RPG). Była prawdopodobnie napisana w BASIC'u a następnie skompilowana stąd jej raczej niska szybkość. Dość dużo czasu zajmują operacje dyskowe (np. samo wczytanie gry zajmuje prawie 5 minut). Tematyka jak zwykle w RPG: potwory, czary, złoto, labirynty a w samym środku my, ktorzy musimy dać sobie z tym wszystkim radę. Ukończenie zajmuje bardzo dużo czasu toteż autor pamiętał o zainstalowaniu opcji zgrywania stanu gry na dysk.



BOXING CHAMPION (c) SIMULMONDO

Kolejna symulacja gry w boks. Tym razem firma Simulmondo zaprosiła do współpracy ludzi za sceny (a ściślej GUY'a SHAVITT'a/THE FORCE jako muzyka oraz ZAGOR'a i ZORIS'a/FLASH INC. jako grafików) co zaowocowało świetną (a



jakże by inaczej) muzyką i grafiką. Można tylko żałować, że nie poproszono jakiegoś kodera z dobrej grupy o wykonanie kodu gdyż wtedy cała gra wyglądała by jeszcze lepiej i na pewno stała by się sporym hitem. Ale i tak grze tej niewiele można zarzucić (może tylko nieco zbyt małą ilość ciosów). Mamy możliwość wyboru przeciwnika. Może to

być nasz kolega lub jeden z kilku graczy (o różnych możliwościach) symulowanych przez komputer. Istnieje też możliwość zapisania stanu na dysk.

STREET FIGHTER 2 (c) US GOLD

Ta bardzo popularna gra z automatów i konsoli doczekała się wreszcie konwersji na C64. Tym razem jednak autorzy gry nie popisali się zbytnio. Kiepska muzyka, małe i brzydko animowane postacie, niedopracowanie gry (np. używanie ROM-owych fontów w menu (!) i sporo błędów powoduje, że gra raczej odstrasza niż przyciąga. Jedyną ładnie wykonaną rzeczą są tła na których walczą przeciwnicy. Tym razem grafik nieźle się postarał. Chociaż w opinii zaprzyjaźnionych graczy, gra nie jest taka zła i "da się w nią trochę pograć", to jednak ja (cały czas będąc pod wrażeniem wersji z automatów) zdecydowanie ją odradzam. Pomysł walk ulicznych można było wykonać o wiele lepiej i ciekawiej.

Krzysztof "BRUSH" Dąbrowski



Cheaty i wskazówki do gier



Te pomocne informacje zostały sprawdzone i nadesłane przez Piotra Andersa (z Danii), znanego wśród użytkowników sieci Earn jako Al-Cohol.

Ivanhoe Gdy gra się zacznie, naciśnij "P" i napisz "ZOBINETTE", następnie naciśnij jeszcze raz "P". Teraz możesz już przeskoczyć do następnego etapu za pomocą klawisza "N".

Dragon Breed Naciśnij Pause i napisz "IREM", co powinno dać Ci wieczne życie. Gdy naciśniesz "N", będziesz mógł przeskoczyć jeden level.

Time Machine Jak sie ukaże High Score, napisz "Dizzy" zamiast swego nazwiska, by potem cieszyć się wiecznym życiem. Spróbuj też nacisnąć "A" i "S".

The Godfather Naciśnij Pause i napisz "PIZZA HUT". Po powrocie do gry będzisz miał nieskończoną ilość energii.

Venus the Flytrap Kody do Leveli:

1. MANTIDS
2. CICADAS
3. PSYLLIDS
4. PIERIDS
5. SATYRID
6. LYCAENID
7. PYRALID
8. NOCTUID

Alien Breed Te kody należy wpisać, kiedy nasz bohater odnajdzie komputer i go uruchomi.

I CANT BE ARSED TO PLAY THE FIRST LEVEL - przeniesie Cię na level 3.
ALIENS ARE FAGGOTS - wrogowie będą słabsi.
PUFFNUTS MODE - wrogowie będą wolniejsi.
IS IT TRUE THAT ALIENS SUPPORT MAN UTD - wrogowie będą bardzo słabi.
PISSED AS A FART - będziesz grał jak pijany.
BEN JOHNSON TRAINED THESE ALIENS - wrogowie będą szybsi.
THE IRAQUIS MADE THE WEAPONS - twoja broń będzie mocniejsza.
SALMAN RUSHDIE PLAYS ALIEN BREED - niespodzianka.
BEWARE ALIENS SPADGE HAS DROPPED ONE - wrogowie będą uciekać.

Spróbuj też: READ, GAMESX, SPADGE, LEEDS, LIVERPOOL, NEWCASTLE, BROWN, GALLUP.

Night Breed Na początku gry musisz jechać do szpitala (prawy dolny róg na mapie). W nocy musisz uciec ze szpitala i jechać na cmentarz (prawy górny róg na mapie).

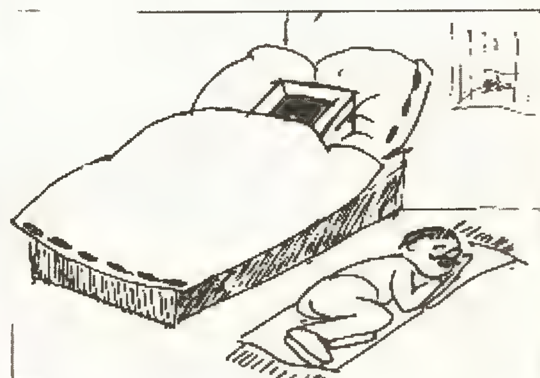
Tu będziesz mógł wybrać między: a) sprawdzaniem terenu. b) odwiedzeniem cmentarza. c) bezpośrednią jazdą do podziemnego świata Midians. Wybierz taką drogę na cmentarzu, aby być szybko napadniętym przez potwora Peloqin. Gdy zostaniesz ugryziony, musisz uciekać (pomaga w tym wciskanie spacji).

Po tym wszystkim spotkasz policję, która pomyśli, że popełniłeś morderstwo. Policja od razu cię zastrzeli, ale po ugryzienu przez potwora stałeś się nieśmiertelny. Powróć na cmentarz i wybierz komendę "Look Around", teraz spotkasz się z rzucającym nożami potworem. Unikaj noży i ruszaj dalej w dal podziemnego świata Midians. Tam spotkasz między innymi Lylę, która da Ci różne informacje. Z resztą gry musisz się już sam pomęczyć.

Alien Breed '92 Special Edition 2. XXDFA 3. RTHAA 4. LAEEA 5. PPEAB

Robocop III Przytrzymaj klawisz Shift i napisz "THE DIDDY MAN". Użyj teraz Esc do przeskakiwania z levelu na level.

Piotr "Al-Cohol" Anders.



COMMODORE

OGŁOSZENIA DROBNE

Sprzedam C64 II, stacja dysków 1541 II, FINAL III, joystick, dyski, literatura. Cena 3.8 mln.
Bartosz Gołębiowski, Gliwice, ul. Puszkina 2/2, tel. 382-596

Programy - Amiga (koperta+znaczek 2.5 tys.)
Henryk Bagiński, Żary
ul. B. Głowackiego 2.

Sprzedam gry i programy na C64 (dysk, kaseta). Informacje: koperta+znaczek
Krzysztof Oborycki, 18-421 Piątnica, ul. Zielona 10.

Skontaktuję się z posiadaczem gry Kolateta na C64. W zamian oferuję inne gry i użytki na taśmie. Kontakt - koperta+znaczek.
Tomasz Pająk, 58-300 Wałbrzych, ul. Kasztanowa 11/3.

Wymiana gier (tylko nowości - dysk) C64 i A500.
Paweł Witek, 58-506 Jelenia Góra, ul. Karłowicza 45/55.

C64 - gry i użytki (taśma/dysk). Opisy. Informacje: koperta+znaczek.
Mariusz Listowski, 76-200 Słupsk, ul. Sobieskiego 17/8.

Klub "X" zrzesza użytkowników C64 i Atari XL/XE. Przyślij swój namiar i typ komputera.
KLUB "X", 59-220 Legnica, ul. Gombrowicza 37/6.

Poszukuję gier na C64 "The Great American Cross" "Street Fighter", "Flimbor's" na kasetach.
C64 C, magnetofon, 25 kaset, 2 joysticki, literatura, Black Box V.3, **zamienię** na Amigę 500.
Szymon Skoczyński, 70-892 Szczecin, ul. Nauczycielska 12

Amiga! Klub korespondencyjny. Gry i użytki gratis. Szczegóły: koperta+znaczek (dopisek: Amiga).
Marcin Mirkowski, 02-761 Warszawa, ul. Maltańska 1/93

Sprzedam: modulator TV do Amigi - nie używany - 500 tys. zł. **Sprzedam** : Amiga 500, 1MB, KICK-ROM 2.04, monitor stereo, color-PHILIPS, zewn. stacja 3.5". Cena 10.35 mln. zł.
Maciej Ba..., 89-500 Tuchola, ul. Główna ...

Poszukuję programu MERLIN-64 lub MACRO ASSEMBLER. Mój adres: Jakub Królikowski, 09-230 Bielsk, Zagroba 23.

C64 - **1000 gier** (kaseta 20 tys. zł) tanio! Katalog gratis, koperta+znaczek.
Piotr Kozłowski, 10-153 Olsztyn, ul. Jeziorna 1/3.

Sprzedaż oprogramowania do Amigi. Duży wybór (nowości), katalog gratis. **Sprzedam** Amiga 500 (gwarancja). Cena do uzgodnienia.
Przemysław Mikosz, 43-300 Bielsko-Biała, ul. ... 7/12, tel. 495-37...

Sprzedam: C64 I, monitor, drukarka ...M + interfejs, FINAL III, X, joy, 27 kaset, literatura. Cena 3.5 mln. zł.
Szymon Grygowski, 72-200 Nowogard, ul. Poniatowskiego 22/27, tel. (0932) 20-481.

Sprzedam płytki drukowane samplera wykonane fotochemicznie + schemat montażowy i układy ADC 0808, ADC 0809.
Wojciech Szewczyk, 32-016 Stanisławowice 303.

Kupię instrukcję po polsku o programie TRANSPUTER MAN 64, cena do uzgodnienia.
Janusz Blacharski, 49-300 Brzeg, ul. Młynarska 3/7.

Sprzedam gry i użytki na C64, tanio!! Informacja koperta+znaczek.
Artur Sikorski, 40-079 Katowice, ul. Gliwicka 2/4

Korespondencyjny Klub Użytkowników Amigi **"ELITE"**. Informacja: koperta+znaczek.
ELITE, 39-400 Tarnobrzeg, ul. Orzeszkowej 3/12.

Sprzedam A500 wersja 6A, 1MB RAM, bootselector przełącznik 1MB Chip RAM. Cena 5 mln zł.
Jacek Nowak, 80-520 Gdańsk-Brzeźno, ul. Pułaskiego 19/21 Em2.

Grupa **"B-TECHDESIGN"** C64 poszukuje członków wszelkich specjalności, a szczególnie koderów i muzyków. Napiszcie!
Adam Kaźmierski, 78-200 Białogard, ul. Kusocińskiego 3/...

EXILES (C64) poszukuje kontaktów oraz członków wszelkich specjalności!
Przemysław Plewa, 85-124 Bydgoszcz, ul. M. Konopnickiej 6/78.

Kupię sprawne ZX-81.
Marian Gabrowski, 58-405 Krzeszów, Lipienica 48.

Sprzedam: Bajtki, C&A, Amigi, TOPSECRET'y oraz rozszerzenie pamięci (1MB), pokrywa itp do Amigi.
Marcin Mirkowski, 02-761 Warszawa, ul. Maltańska 1/93

Kupię oryginalną wersję TURBO ASSEMBLERA V5.1 (na dyskietce).
Krzysztof Lotko, 45-761 Opole, ul. Koszyka 30/32.

Sprzedam: C64 II + magnetofon 1530 (gwarancja) + stacja dysków 1541 II (gwar. do XI 93) z bardzo bogatym osprzętem. Stan całości idealny. Oferty listowe+cena. Więcej informacji - listownie.
Robert Owczarek, 55-100 Trzebnica, ul. Polna 10/11.

UWAGA! Czy chcesz napisać własne demo? Nic prostrzego. Napisz a sam się przekonasz (C64).
Rafał Tylus, 62-400 Słupsk, ul. Niepodległości 16/36.

Sprzedam A500 2, 6MB RAM, ACTION REPLAY MK III, 100 dysków, literatura, inne. Cena 8.3 mln zł.
Piotr Krzemiński, Ostrowiec Św., Os. Putanki 14/10, tel. 626-265.

Sprzedam Flicker Fixer "Multivision 500" i 1MB pamięci do 500 plus.
Grzegorz Drabek, 71-471 Szczecin, ul. Wiosny Ludów 1/4 tel. 536-235.

"KOŁO FORTUNY" na C64 - kaseta 20 tys + opłata pocztowa
Norbert Marcinkiewicz, 91-961 Łódź, PO Box 149.

Wymiana programów na Amigę !!!.
Tomasz Wagner, 05-220 Zielonka, ul. 11 Listopada 8ᵃ.

Zamienię: C64, 1541 II, 1535, mysz, 2 joy, 60 dysków na 2-letnią Amigę 500.
Krzysztof Puk, 41-704 Ruda Śl. 4, ul. Bogusławskiego 5/43.

Stację dysków 1571 **sprzedam** lub **zamienię** na stację OCEANIC.
Jerzy Tederko, 61-400 Suwałki, ul. Kowalskiego 6/32.

A 500 półtoraroczną niedrogo **sprzedam**, **kupię** C64 (czarna klawiatura).
Dariusz Miszczak, 90-322 Łódź, ul. Kilińskiego 160/30.

Kupię MC 68000 książkę i mapę pamięci Amigi.
Andrzej Sławiński, 59-830 Olszyna, ul. Kamienna 2.

Sprzedam Amigę 500, 1MB + 200 dyskietek do nagrania - ok. 5.5 mln., monitor RGB color - 20" - cena 1.5 mln.
Tomasz Choinacki, 32-300 Olkusz, ul. Powstańców Śl. 57 tel. (0356) 31-571.

Coder, cracker, C64 - **wstąpi** do grupy.
Grzegorz Bąbka, 31-344 Kraków, ul. Ojcowska 124.

Sprzedam A 500, rozszerzenie z zegarem (do 1 MB), joystick, dużo literatury, dyskietki. **TANIO! PILNIE!**
Marek Opyrchał, 30-057 Kraków, ul. Chocimska 20/7, tel. (012) 340-739.

Grupa "DUET" **nawiąże** kontakty w celu wymiany: grami, programami i doświadczeniami, 100 % answer.
Bartek Uptas, 78-100 Kołobrzeg, skr. poczt. 238.

Grupa "ATHEIST" (C64) **poszukuje** koderów i mega swapperów (prężnych i odpowiedzialnych).
M. Cajsel, 92-591 Łódź ul. Gorkiego ...

Pilnie **kupię** oryginał TURBO ASSEMBLER V.5.
Karol Srebrowski, 19-300 Ełk, ul. Dobrzańskiego 5/67.

Wymiana programów użytkowych, gier, dem (dyskietka) C64.
Krzysztof Świtajski, 87-100 Toruń, ul. Buszczyńskich 9/34.

Sprzedam roczną A 500 + rozszerzenie do 1MB + 30 dysków + literatura. Cena ok. 6 mln., **sprzedam** TOP SECRET (1-7), Bajtki: 3-4 i 9/86, 3, 5, 9/87, 2, 7/88, 1, 4, 5/89, 5-6, 11-12/90, 1, 2/91 w cenie 10 tys. za egzemplarz.
Sebastian Budzis, 76-200 Słupsk, ul. A. Struga 6, tel. 440-234.

Dwóch średnio zaawansowanych grafików (C64) **szuka kontaktu** z grupą komputerową.
Michał Rodak, 64-200 Wolsztyn, ul. Żeromskiego 26/10.

Sprzedam: C64 II, 1541 II, FINAL II, monitor TWM-315, 1531, 60 dysków, 15 kaset, 3 joysticki, literatura, cena 4 mln. zł.
Piotr Żarczyński, 14-100 Ostróda, ul. Rycerska 6/22.

Sprzedam ponad 1000 programów (kaseta) na C64. Katalog: koperta + znaczek.
Tomasz Dąbkowski, Racław 15, 66-432 Baczyna

```
;-----------------------
; info - scroller   autor:Pawel Soltysinski
; Commodore Kebab 1993
;-----------------------
          *= $1000              ; adres startu

          SEI                   ;  wylaczenie przerwan
          LDA #$7F              ;  USTALANIE PARAMETROW PRZERWAN:
          STA $DC0D             ;
          LDX #$00              ; wylaczanie Timer'˘w
          STX $DC0E             ;
          INX                   ;
          STX $D01A             ; ustawianie przerwan rastrowych
          LDA #$1B              ;
          STA $D011             ;
          LDA #$E0              ; ustalanie numeru linii do przerwania ($ED)
          STA $D012             ;
          LDA #<IRQ
          LDY #>IRQ
          STA $0314             ; ustawianie nowego wektora IRQ
          STY $0315             ;
          LDA #<TEXT
          STA $FB               ; ustawianie wektora poczatku tekstu
          LDA #>TEXT
          STA $FC
          LDX #$27
          LDA #$20
          STA $0370,X           ; kasowanie obszaru dla scroll'a
          DEX
          BPL *-4
          CLI
          RTS

IRQ       LDA $DC01             ; testowanie - wcisnieto CTRL
          CMP #$FB
          BEQ IRQ2              ; wcisnieto - id§ do IRQ2
          INC $D019             ; nie wcisnieto - koncz przerwanie
          JMP $EA31
IRQ2      LDX #$27              ; przepisz ostatni wiersz ekranu do pamieci
          LDA $07C0,X
          STA $0340,X
          LDA $0370,X           ; pobierz i wpisz do ostatniej linii ekranu
          STA $07C0,X           ; tekst scroll'a
          DEX
          BPL IRQ2+2
          LDA #$F2              ; odczekaj, az rozpocznie sie wyswietlanie linii
          CMP $D012             ; o numerze $F2
          BNE *-3
          LDX #$09
          DEX                   ; dopasowujaca petla opozniajaca
         BNE *-1
SCROLL    LDA #$07              ; wstawianie odpowiedniej wartosci do $D016
          STA $D016
          LDA #$FD              ; odczekanie, az zakonczy sie wyswietlanie
          CMP $D012             ; obszaru tekstowego ekranu
          BNE *-3
          LDX #$27
IRQ3      LDA $0340,X           ; odtworzenie pierwotnego wygladu ostatniego
          STA $07C0,X           ; wiersza ekranu
          DEX
          BPL IRQ3
          LDA #$C8              ; ...i wstawienie standardowej wartosci do
          STA $D016             ; komorki $D016
          JSR DOTEXT            ; wywolanie procedury scroll'a
          INC $D019
          JMP $EA7E             ;  zakonczenie przerwan

DOTEXT    DEC SCROLL+1          ; przesuniecie w lewo o jeden bit
          BMI *+3               ; wynikiem jest $FF? Tak - kontynuuj
          RTS                   ; wynik rozny od $FF - powrot
          LDA #$07
          STA SCROLL+1          ; ustawienie parametru dla $D016 na 7
          LDX #$00
DOTEXT2   LDA $0371,X           ; przestawienie znakow w pamieci o jeden w lewo
          STA $0370,X
          INX
          CPX #$27
          BCC DOTEXT2
          LDY #$00
READ      LDA ($FB),Y           ; pobranie kolejnej litery tekstu
          BNE DOTEXT3           ; nie koniec - idz do DOTEXT3
          LDA #<TEXT            ; ponowne ustawienie poczatku tekstu
          STA $FB
          LDA #>TEXT
          STA $FC
          JMP READ              ; ...i ponowne pobranie litery
DOTEXT3   AND #$3F              ; przeliczenie z ASCII do kodow ekranowych
          STA $0397             ; wstawienie nowego znaku z prawej strony
          INC $FB
          BNE *+4               ; zwiekszenie wektora tekstu o jeden adres
          INC $FC
          RTS

TEXT      .TEXT "CZYTAJCIE COMMODORE KEBAB"     ; tekst do wyswietlenia
          .TEXT " - PISMO DLA WSZYSTKICH"
          .TEXT " POSIADACZY C-64 I AMIGI!"
          .TEXT "                 "
          .BYTE $00                              ; znacznik konca tekstu
```

"KOLO FORTUNY "

$0801-$13CF

```
:0801 0B 08 90 06 9E 32 30 34 (1E)
:0809 39 00 A0 00 78 E6 01 B9 (B5)
:0811 20 13 99 FA 00 C8 D0 F7 (2A)
:0819 4C 00 01 A7 41 55 54 92 (2B)
:0821 3A 20 50 69 86 58 39 53 (0C)
:0829 7E 54 59 53 49 5C 53 4B (E0)
:0831 49 B1 B6 0A 83 20 43 4F (CA)
:0839 4D 4D AB 0B 96 61 C6 4B (4B)
:0841 45 42 41 42 20 4C 54 44 (B1)
:0849 00 72 F3 17 83 83 0B FD (40)
:0851 8E 33 74 4E 0A 57 66 F8 (A7)
:0859 85 A5 15 0C 20 20 6C 61 (FB)
:0861 53 8F 7D D0 02 D1 59 FC (D0)
:0869 BE 98 0E 8C D7 05 45 77 (A5)
:0871 EA F3 5C 57 59 42 CA BA (58)
:0879 E7 F5 30 50 C7 5F 20 4B (77)
:0881 4C 41 F8 6A F6 53 93 4D (14)
:0889 49 3C 52 E0 20 20 44 4F (7C)
:0891 FA 4C 00 FB 2F 5E E4 21 (7A)
:0899 46 4F 52 D3 82 4E 59 C7 (CC)
:08A1 A5 8D FC 8C 1E 1B 5A 16 (EA)
:08A9 8D D0 52 86 07 CE B0 05 (DB)
:08B1 CD 2C 09 F0 02 54 56 DB (ED)
:08B9 E8 C8 EC 89 90 DC B9 6D (90)
:08C1 D2 61 F2 C8 C0 52 D3 88 (04)
:08C9 14 9F 1A 73 F8 6E B4 1C (75)
:08D1 76 69 00 A6 13 18 51 BB (B7)
:08D9 D5 88 54 0E C9 02 90 03 (FB)
:08E1 EB 3A 0E 60 00 ED 2C 7E (A4)
:08E9 D6 86 FF 22 80 A9 CF 96 (27)
:08F1 53 42 4A 2C 44 32 30 84 (4E)
:08F9 35 C9 63 F3 C9 03 A6 97 (FE)
:0901 68 68 4C 3C 10 EB 02 19 (BE)
:0909 66 0E E2 27 40 8E 90 09 (A2)
:0911 A2 49 B5 12 6C 70 5C 14 (95)
:0919 D0 08 E0 9C CA C5 CA 4C (88)
:0921 98 09 EC 8F 09 B0 BC 9A (15)
:0929 3B 83 B7 E9 26 10 01 9D (49)
:0931 49 07 E8 D0 A2 8C 36 7A (45)
:0939 8B 17 27 17 2B 08 93 08 (18)
:0941 F0 0F 62 F9 20 A9 04 26 (44)
:0949 53 D3 C8 98 2C 48 26 2C (F9)
:0951 22 8B A2 D4 40 E4 09 40 (9F)
:0959 44 F4 14 A6 3C D8 21 F0 (05)
:0961 7B 04 01 2C A9 0B 32 E5 (B5)
:0969 97 A1 A0 28 50 C8 91 20 (02)
:0971 A5 21 58 37 D4 85 21 3F (66)
:0979 25 02 B4 04 C6 3B 0F EA (D0)
:0981 6D 4F A5 F3 C9 69 28 85 (F3)
:0989 20 A7 D6 E6 21 22 83 9E (10)
:0991 60 85 22 26 0A 85 23 61 (4F)
:0999 BC 64 D0 0C 08 B4 C5 22 (99)
:09A1 E0 AF 9B 5D E6 23 70 92 (1D)
:09A9 B6 5E 2C C9 27 F0 16 DA (99)
:09B1 B2 F6 18 60 38 D0 78 18 (20)
:09B9 47 46 EC A0 28 AD 66 8C (D9)
:09C1 7F 96 FC 23 5A 5C 88 81 (9F)
:09C9 6C 18 90 02 A0 D2 A5 37 (6D)
:09D1 EF 8C 10 D0 C9 B0 00 0A (AE)
:09D9 F0 B3 32 8D 01 67 98 0B (F1)
:09E1 0B A9 92 53 4F 31 EA A2 (70)
:09E9 96 61 D0 FD A9 1A 8D 18 (32)
:09F1 D0 A9 A8 8D 14 DE 25 0A (33)
:09F9 FE E3 02 EC C6 12 D0 EE (E6)
:0A01 19 D0 4C 7E EA B5 FB F0 (CD)
:0A09 04 CE 66 60 71 1B 8D 4E (87)
:0A11 60 28 0B 8D 27 D0 E8 8A (6B)
:0A19 29 07 8D 19 0B 60 06 0B (5E)
:0A21 0C 08 0C 0B 06 00 C9 B0 (B4)
:0A29 19 FD 1E 5D D0 4F C9 BC (5D)
:0A31 A8 07 5E D0 47 C9 B1 EA (8B)
:0A39 5F D0 3F C9 B6 81 7A 5B (E5)
:0A41 D0 37 C9 AA A0 1E 5C D0 (64)
:0A49 2F C9 B9 A8 07 23 D0 27 (BC)
:0A51 C9 AE EA 24 D0 1F C9 AD (7F)
:0A59 81 7A 25 D0 17 C9 BD D0 (5B)
:0A61 04 A9 26 D0 0F A0 BB E3 (B3)
:0A69 F0 0B C9 41 B0 03 D8 40 (42)
:0A71 BB 5B B0 36 86 FF 60 93 (A4)
:0A79 08 39 05 88 A3 45 44 34 (75)
:0A81 43 4A 58 84 54 20 53 45 (4B)
:0A89 F1 BD 52 4E 68 A5 20 59 (BA)
:0A91 4B 4F 5C 33 15 5E 00 04 (21)
:0A99 2D 17 52 3B E2 5A 7F 5C (BF)
:0AA1 C0 2E 0D 76 04 9C 86 20 (2C)
:0AA9 BF 45 DA D1 75 01 32 F7 (33)
:0AB1 16 80 44 5A 45 4E 11 D5 (51)
:0AB9 12 33 B7 16 32 41 54 31 (0C)
:0AC1 02 34 EE 4C DD 54 55 5B (A3)
:0AC9 9E 16 70 35 7B 5A 72 9D (4A)
:0AD1 E8 31 57 49 45 49 01 36 (14)
:0AD9 37 27 90 4E 4E 4F 24 5E (9C)
:0AE1 A5 0A B9 7E D5 37 2D 20 (75)
:0AE9 4F 53 4F 42 41 0D 00 E0 (70)
:0AF1 0B F2 0B 0A 0C 1C 0C 2E (DB)
:0AF9 0C 45 0C 5A 0C A9 8A FA (ED)
:0B01 AB 66 49 8A 7A 82 BC D8 (D8)
:0B09 B3 1E 50 D0 92 B6 A2 3F (B7)
:0B11 18 E2 2B 2E 01 09 B0 8E (AC)
:0B19 3F 8E DD 8D 25 F7 9D 40 (18)
:0B21 03 9D 41 03 58 20 AB 2D (C5)
:0B29 90 F3 4F 38 8D F8 07 4F (B1)
:0B31 30 85 CF F5 1B 3B 19 50 (CF)
:0B39 21 F0 FB 82 2A 12 CB 45 (31)
:0B41 06 BF 05 74 82 F8 30 EB (B1)
:0B49 F3 45 15 D0 60 C9 11 EA (AD)
:0B51 07 C9 05 B0 DA EE C6 EA (54)
:0B59 C0 91 FE 40 CB CE 0A A4 (71)
:0B61 0A 42 1D D0 0D 09 C9 13 (1F)
:0B69 B0 BA EE A1 3A C9 9D D0 (89)
:0B71 0B AD AA 12 F0 AB CE 13 (13)
:0B79 4C BF 28 66 30 0B F0 A0 (20)
:0B81 A5 6D 46 4D 08 F0 06 6A (53)
:0B89 FA 14 CA D0 FA 18 6D 4C (21)
:0B91 08 AA 68 C9 E6 6E B6 24 (80)
:0B99 00 DF 87 6C C9 00 B1 6F (E3)
:0BA1 33 F5 4C FE 0C 20 6F 0C (0A)
:0BA9 C0 D6 2A 14 37 69 BF A0 (B0)
:0BB1 0B 96 4D 1A 38 EE 26 C5 (20)
:0BB9 98 60 78 A2 FB 9A BB 12 (3C)
:0BC1 E4 CD 99 BF E3 23 40 D6 (C2)
:0BC9 A2 77 B0 44 63 42 49 2A (4E)
:0BD1 CE 0C 68 A9 4E D0 50 A9 (7C)
:0BD9 9C A0 99 E1 02 14 C6 96 (AB)
:0BE1 BD E5 23 0D 9D 2C 2C 7F (55)
:0BE9 5A F5 79 1C 25 6E 82 83 (06)
:0BF1 66 06 85 0E C9 33 28 38 (2C)
:0BF9 18 11 4C 41 12 A6 BA 67 (B2)
:0C01 71 0D 5F C0 54 45 21 A5 (06)
:0C09 50 52 23 42 4A 5A BC 80 (2C)
:0C11 D9 70 86 E8 FC 01 68 AA (22)
:0C19 2D 24 50 73 8A A7 28 78 (CA)
:0C21 BD FC B9 99 4F E8 C8 C0 (E4)
:0C29 03 90 F4 AD E9 07 AE E8 (A1)
:0C31 07 7A 3B 30 12 85 D3 65 (0E)
:0C39 04 EA 07 4C 26 25 9F EB (AF)
:0C41 82 93 88 4C 49 31 3C 42 (04)
:0C49 5F A2 B3 62 04 31 2D 33 (A6)
:0C51 00 9E B2 6A 06 55 59 50 (62)
:0C59 0B 4F B5 41 2E B1 7B 87 (D2)
:0C61 F1 0A 72 9B B8 3D C1 0E (F1)
:0C69 D1 03 A3 88 4D DC 45 A5 (09)
:0C71 77 31 90 F9 C9 34 B0 F5 (87)
:0C79 38 E9 31 8D A9 2B 65 74 (08)
:0C81 2E 70 0B 39 D8 6E A0 0E (3C)
:0C89 96 C1 78 2D 08 FF BD 61 (1E)
:0C91 45 A0 BD 62 0C 85 A1 22 (B2)
:0C99 96 C2 8C 5F B1 A0 45 09 (3F)
:0CA1 C8 2B D0 F6 20 F6 3E 9D (11)
:0CA9 D7 B4 CC A9 B1 A7 79 70 (2A)
:0CB1 65 12 7D E5 F3 A0 86 9F (72)
:0CB9 AE 3C 3A F7 81 CC AC 3E (66)
:0CC1 36 90 A4 A8 AD EA 41 20 (53)
:0CC9 7A 0A 9F E0 0F C1 28 90 (29)
:0CD1 55 B9 F0 3B 2D 4A 4B A7 (42)
:0CD9 1A 3C A9 64 4A 91 09 47 (51)
:0CE1 D4 E0 01 F0 CA 20 2D 09 (79)
:0CE9 B0 11 65 AB 6A 5F F0 01 (86)
:0CF1 C8 95 C3 F5 B0 18 20 3C (CC)
:0CF9 10 08 4B CD 4C 0E 90 05 (22)
:0D01 A1 C6 8D FA EE C2 8F 4C (45)
:0D09 F9 0E A4 23 D5 19 0A 0A (F8)
:0D11 AA 98 18 7D 58 21 F3 C5 (7F)
:0D19 D6 FE DD 0D FE DE 57 A9 (96)
:0D21 61 13 BD 37 D0 A5 0D 21 (19)
:0D29 BC 2C E5 96 78 EF 20 09 (6B)
:0D31 0A A2 48 7A 7D AA DB 07 (EE)
:0D39 B3 E8 90 F8 7A A6 98 A2 (D7)
:0D41 14 01 EC D3 62 A6 1B B4 (9F)
:0D49 99 AD DF 0D 56 E2 31 3B (43)
:0D51 81 24 45 A9 21 44 30 25 (4F)
:0D59 02 75 20 7C 10 60 05 75 (FD)
:0D61 CC 0D 9A 18 1B 47 41 44 (9A)
:0D69 5B 20 AF E8 0A 1A DB 12 (19)
:0D71 F3 55 98 A9 2D 8D 02 A0 (C4)
:0D79 03 8D 03 C4 2C 46 84 25 (00)
:0D81 8C 41 16 46 07 A0 FF 8C (32)
:0D89 74 8A E9 40 46 D4 51 5F (5E)
:0D91 10 FF 20 3A 16 EB 8D E8 (FF)
:0D99 E5 B1 A9 08 08 3B A9 13 (C9)
:0DA1 20 8D 00 D4 8D 01 D3 4A (14)
:0DA9 09 8D 05 4C A2 04 06 4F (FC)
:0DB1 0F 8D 44 A0 A9 11 8D 04 (E1)
:0DB9 D4 A2 F0 59 46 C6 02 D0 (A2)
:0DC1 FC A9 10 85 02 88 D0 FB (22)
:0DC9 CA D0 F8 8C 18 D4 60 D7 (20)
:0DD1 17 B0 20 45 61 54 55 4A (49)
```

```
:0DD9 45 55 7A A5 4C 45 85 97 (4C)
:0DE1 54 2D 4E 00 A6 C5 16 89 (44)
:0DE9 D8 59 57 18 A4 84 81 22 (C8)
:0DF1 25 BE 56 A4 64 CF 84 B4 (3B)
:0DF9 0C E9 06 B0 11 35 42 59 (DF)
:0E01 22 16 48 D8 D1 5B 4F 38 (B5)
:0E09 52 E5 51 8D 1D BA 4D 0D (CA)
:0E11 C0 15 8D 30 11 B1 2C 54 (BF)
:0E19 2D DA C0 30 2F 54 EE 16 (1D)
:0E21 A2 C9 6F B0 3D 10 3A 8D (FF)
:0E29 86 02 6F 52 29 A2 0D 8A (9A)
:0E31 69 01 92 A9 FC 82 96 52 (A6)
:0E39 AE 75 7D 20 CD BD 33 C5 (D2)
:0E41 B5 6D A3 4E F0 07 2D 9B (EC)
:0E49 F5 4C 1D 11 A0 45 A6 D3 (5F)
:0E51 A0 10 CB 22 D3 C1 A4 64 (49)
:0E59 47 08 FE C8 A0 3B 95 89 (B5)
:0E61 48 D0 9E B2 08 89 1D F1 (AA)
:0E69 23 FF 85 FB A9 2F 85 FC (FD)
:0E71 AD 69 F1 2C 4B A8 A9 FB (5F)
:0E79 D2 D8 FF 03 C2 08 0D 50 (E7)
:0E81 4F 60 16 4A 20 4E 41 5A (13)
:0E89 57 45 3A 00 A9 DD A0 11 (89)
:0E91 0B DF 72 E0 20 CF 31 AA (5F)
:0E99 0D F0 08 9D 10 01 E8 E0 (CE)
:0EA1 10 90 F1 8E EB B5 A9 0D (C6)
:0EA9 44 01 AE A9 80 85 9D AE (04)
:0EB1 51 D9 60 93 05 57 47 52 (D2)
:0EB9 C5 4E 32 44 41 4E 45 20 (CA)
:0EC1 5A 20 54 41 53 4D 59 20 (45)
:0EC9 43 5A 59 CE B7 59 53 4B (57)
:0ED1 55 20 28 54 2F 44 29 3F (D6)
:0ED9 20 5D 6F 83 78 20 A3 53 (3F)
:0EE1 15 FD 20 18 48 9F 20 D0 (40)
:0EE9 8D 21 D0 A9 18 A0 91 F6 (B9)
:0EF1 1E AB 20 E4 59 AA 04 C9 (80)
:0EF9 44 D0 F5 20 D2 FF C9 54 (7D)
:0F01 F0 03 A2 08 2C A2 01 A9 (03)
:0F09 01 A8 20 BA FF 20 ED 11 (6F)
:0F11 AD 90 09 A2 10 A0 01 20 (A7)
:0F19 BD FF 20 F1 9D 20 D5 FF (A3)
:0F21 90 03 AC 5C A2 0B A0 9D (EE)
:0F29 DC 0D CA 10 FA 20 44 E5 (72)
:0F31 A9 00 8D A4 17 20 1D 08 (5E)
:0F39 20 9B 0E 12 9B 18 69 30 (06)
:0F41 8D B3 12 A2 00 BD 00 30 (EF)
:0F49 9D 00 2D E8 D0 F7 20 C1 (DE)
:0F51 0E EE 01 BF AD 40 12 CD (10)
:0F59 FF 2F 90 DE 4C 73 0D C2 (86)
:0F61 2E E0 3F 3B 80 FF EC 00 (F5)
:0F69 FE B3 03 F8 CF 0E E0 3F (3C)
:0F71 3B 80 FF EC 00 84 B2 F3 (F6)
:0F79 27 BE CC BC 5B E7 DA 38 (66)
:0F81 CF CC B9 68 23 4F 65 F2 (9E)
:0F89 2F 20 FB 80 39 2A C7 2E (F2)
:0F91 B4 81 4D FE 55 DE 81 F3 (31)
:0F99 30 E7 1C 38 CF 32 4F 98 (FA)
:0FA1 7D 8C C3 3A 57 98 2E 47 (33)
:0FA9 23 1E 48 12 4E 5F D3 7C (9C)
:0FB1 23 40 2D 4F 2E F3 2C 4E (62)
:0FB9 00 76 2D FE 27 E0 07 39 (2F)
:0FC1 DA 3C 91 09 8C 7F 5C 2A (83)
:0FC9 EE 37 37 87 A2 F6 55 28 (76)
:0FD1 43 25 4E 75 7E 71 2F 4F (08)
:0FD9 00 8F 27 1F 2D 4D C0 21 (EE)
:0FE1 E2 38 AE 1F 20 D7 FF 8A (BB)
:0FE9 8C 9D 2F 37 0F 32 FA C0 (74)
:0FF1 F2 21 C7 E2 F1 08 67 32 (57)
:0FF9 9E D4 9C C3 5C 8A B1 98 (CD)
:1001 3A 90 08 4E C4 E2 F1 A9 (BA)
:1009 4C 8E 95 23 3A EC A9 E2 (25)
:1011 CC 57 6A 0E 3A 79 FA D9 (A7)
:1019 D0 67 98 77 9B C7 38 07 (DC)
:1021 7C CA 35 72 85 39 50 B3 (5F)
:1029 8F 71 28 C7 5D 8F 19 9A (E8)
:1031 C2 43 3A F1 E1 81 44 39 (0A)
:1039 63 73 5D C3 7C 81 82 2B (0D)
:1041 BC DA 59 1A 99 DA 54 7D (81)
:1049 90 25 0C AF 8E 48 31 3C (C0)
:1051 F2 F1 02 91 CA EF 1F 97 (9C)
:1059 AA FB 5D C1 01 B2 D9 E5 (6C)
:1061 FA 02 CD 01 BA D9 02 F1 (28)
:1069 FA B1 0F FA F9 F0 F2 55 (AD)
:1071 0A C1 BA 79 88 DD C3 55 (F2)
:1079 AA 60 FD 02 D6 FE 8A 8C (3A)
:1081 9D 67 E8 5C 49 CE 15 3A (C8)
:1089 17 72 AE 23 F1 51 3B F7 (1A)
:1091 EA 39 47 4D EE A7 CC 81 (32)
:1099 9A 23 CD 31 4E 27 43 2D (61)
:10A1 2F 4D 2F 4F 2E 45 0F 4E (A0)
:10A9 F4 27 13 41 21 C4 6B E2 (72)
:10B1 7C 34 07 70 8A 0E 99 A3 (C7)
:10B9 F2 C1 51 F9 D7 4D 80 3C (75)
:10C1 85 4C 2C 4F 27 A3 17 CC (44)
:10C9 81 43 27 4F 2F 1C 3D 90 (4F)
:10D1 00 20 40 2A F9 AF CA D8 (C6)
:10D9 B9 D2 3C C2 F9 73 9C 2B (2D)
:10E1 CD 33 F9 ED 2F 37 FF E2 (01)
:10E9 B6 8F C4 01 D6 03 73 4F (FA)
:10F1 F3 0F 87 F2 0E 9C 7B 99 (82)
:10F9 1B 69 7A B9 DA 33 FA F2 (A2)
:1101 71 28 3C 2A 0F CD 91 E6 (6F)
:1109 9D 39 72 E1 96 97 D9 FA (3A)
:1111 79 3A A2 07 82 14 53 07 (90)
:1119 12 C1 C2 C4 6F E4 3C C3 (53)
:1121 F9 7A 39 1A E8 15 19 18 (A7)
:1129 0D 96 0F 38 EB 08 01 62 (5E)
:1131 43 AB 19 A2 F8 00 27 50 (17)
:1139 9E C2 39 68 19 1A F9 F2 (2F)
:1141 F4 20 81 02 F9 FA 8E 1E (9C)
:1149 48 02 01 AA 55 00 0A 48 (80)
:1151 C6 CE 97 99 77 E5 51 8C (35)
:1159 BA 12 93 45 B6 3A 64 3E (AB)
:1161 E4 48 3C 23 E6 C2 C2 A5 (A6)
:1169 18 91 6B 99 0B 7E 03 06 (C9)
:1171 BF 86 B9 C2 00 C8 3B 02 (DD)
:1179 60 79 46 DE 6D 9E 30 0F (C3)
:1181 B8 C3 37 F2 21 97 16 F4 (A6)
:1189 A1 38 C2 81 1B 00 79 64 (EB)
:1191 AE 24 07 3C FA 5A 29 A2 (CA)
:1199 EA 40 AC 91 2B 33 D7 CA (96)
:11A1 37 4C 70 D2 3D 59 18 66 (38)
:11A9 76 7E 6E 5B 97 C8 7E D5 (9F)
:11B1 9D 03 D8 64 EA 60 3E 40 (01)
:11B9 7C B1 60 68 7E 03 A8 D5 (30)
:11C1 06 80 A0 21 11 69 A8 EC (FF)
:11C9 18 8A 38 18 C7 11 3C FE (EB)
:11D1 5E 7E 0C 7C E5 81 0C 7E (13)
:11D9 2C 18 E8 B7 02 E0 0E B1 (0E)
:11E1 D2 AD 63 6E 76 2C 3E 18 (C7)
:11E9 38 11 02 18 3F 1C 30 60 (ED)
:11F1 7E 2A FA 1E A1 1F 45 D3 (94)
:11F9 11 2E 7E 7C F4 3E 60 7C (99)
:1201 A9 7C 7C D4 00 06 FA 40 (72)
:1209 8A 66 3C 78 3C 66 3E 06 (77)
:1211 06 7C C4 1A 20 18 60 00 (A5)
:1219 00 0C 06 03 03 06 0C BC (C8)
:1221 2C 20 00 7E C8 C8 C8 A9 (EF)
:1229 00 85 10 A6 FB 30 0D 06 (9F)
:1231 FA 2A 26 10 C6 FB 88 D0 (3B)
:1239 F2 AA 18 60 48 A1 FF 85 (A8)
:1241 FA A2 07 86 FB E4 FF A6 (26)
:1249 FE D0 02 C6 FF C6 FE 68 (E8)
:1251 90 DD E0 E7 D0 D9 A9 37 (66)
:1259 85 01 58 4C 10 08 A9 10 (C9)
:1261 E6 11 24 11 85 04 AA BC (62)
:1269 D8 07 20 24 07 A6 04 7D (5C)
:1271 BA 07 48 A5 10 7D C7 07 (9E)
:1279 A8 68 A6 02 D0 08 C0 00 (7D)
:1281 D0 04 C9 03 F0 D8 18 65 (62)
:1289 FC AA 98 65 FD 85 49 A4 (75)
:1291 11 F0 20 8A 38 E5 11 B0 (89)
:1299 03 C6 49 38 85 48 A5 FC (A1)
:12A1 E5 11 B0 02 C6 FD 85 FC (21)
:12A9 B1 48 88 91 FC D0 F9 C4 (93)
:12B1 12 F0 88 B1 48 C6 FD C6 (38)
:12B9 49 C6 12 10 ED 00 08 19 (B7)
:12C1 39 7E FF 7F 7F 00 10 20 (EC)
:12C9 40 80 00 00 00 00 00 00 (1B)
:12D1 01 02 00 00 00 00 00 01 (F0)
:12D9 03 07 0F 03 04 05 06 07 (C9)
:12E1 07 08 08 04 04 05 06 07 (C6)
:12E9 09 0A 0B 01 03 C8 E9 2F (D3)
:12F1 F4 6E 99 D4 AE 52 F4 6D (54)
:12F9 78 A9 7F 8D 0D DC B0 75 (67)
:1301 0E DC E8 8E 1A 99 4A 1B (C0)
:1309 8D 44 4D CE 5B 7E 17 58 (6C)
:1311 DC 04 10 FB 2C 11 D0 30 (96)
:1319 FB B2 42 8A 1D 8C 4E 86 (A4)
:1321 06 00 2A 24 12 B9 20 12 (68)
:1329 99 1C 07 C8 D0 F7 20 23 (14)
:1331 07 F0 46 20 23 07 D0 30 (86)
:1339 20 22 07 69 02 C9 04 90 (C5)
:1341 27 D0 07 20 23 07 69 04 (88)
:1349 D0 1E 20 21 07 69 06 C9 (57)
:1351 0D D0 11 C8 20 21 07 69 (43)
:1359 0D C9 0E D0 07 A0 01 20 (5F)
:1361 24 07 69 24 EE 00 04 EA (83)
:1369 85 11 A6 FE A5 FF 20 8A (70)
:1371 07 A5 49 85 FF A5 48 85 (BD)
:1379 FE 20 23 07 85 02 F0 14 (24)
:1381 0A 2C A9 03 85 11 20 21 (E4)
:1389 07 A6 02 D0 02 69 08 20 (ED)
:1391 61 07 F0 9A 20 23 07 F0 (6E)
:1399 E9 20 22 07 69 04 C9 06 (2B)
:13A1 90 E2 D0 07 20 22 07 69 (79)
:13A9 06 D0 D9 A0 05 20 24 07 (7A)
:13B1 69 0A C9 16 F0 0D C9 18 (31)
:13B9 D0 CA A0 01 20 24 07 69 (05)
:13C1 31 D0 C1 A0 09 20 24 07 (89)
:13C9 A6 10 86 12 90 B6 00 00 (90)
```

```
;--------------------
;program wczytujacy pod dowolny adres
;programy nierelokowalne z tasmy.
;(c) BRUSH/ELYSIUM for commodore KEBAB
;--------------------

        *= $c000

        lda #$00        ;ustawienie parametrow
        jsr $ffbd       ;nazwy pliku
        lda #$01
        tax
        ldy #$ff        ;okreslenie urzadzenia
        jsr $ffba       ;pliku logicznego itp.
        lda #$00
        ldx #$01        ;LSB adresu ladowania
        ldy #$08        ;MSB adresu ladowania
        stx $c3
        sty $c4
        sta $93
        sta $90
        jsr $f817       ;wyswietl. PRESS PLAY
        bcs error       ;detekcja bledu
        jsr $f5af       ;komunikat SEARCHING
        jsr $f72c       ;wczytanie naglowka
        beq error       ;detekcja bledu
        bcs error       ;---"---
        ldy #$01
petla   lda ($b2),y     ;zapamietanie oryginа-
        sta $c0f0,y     ;lnych adresow ladowa-
        iny             ;nia od $c0f0 oraz
        cpy #$05        ;wczytanie pliku pod
        bne petla       ;wybrany adres
        jmp $f57d
        rts             ;wyjscie z programu

error   lda #$00        ;wyjscie z programu po
        sta $d020       ;stwierdzeniu bledu.
        rts
```

"LOADER "

```
$C000-$C040
--------------------

:C000 A9 00 20 BD FF A9 01 AA (05)
:C008 A0 FF 20 BA FF A9 00 A2 (AF)
:C010 01 A0 08 86 C3 84 C4 85 (AC)
:C018 93 85 90 20 17 F8 B0 1A (88)
:C020 20 AF F5 20 2C F7 F0 12 (83)
:C028 B0 10 A0 01 B1 B2 99 F0 (EC)
:C030 C0 C8 C0 05 D0 F6 4C 7D (64)
:C038 F5 60 A9 00 8D 20 D0 60 (D9)
```

Po dużej przerwie jest już do nabycia 3 Kebab-Public-Domain, na Amigę, a na nim między innymi:

- **PC-Emulator 1.63** - zarchiwizowana DMSem, demonstracyjna wersja najszybszego jak na razie programowego emulatora komputerów iBM. Programowa emulacja karty EGA oraz chluby każdego iBMowca, karty dzięko-podobnej COVOX, stawia tem program w czołówce emulatorów tego typu. Program może być rozpowszechniany przez każdego (jest to bowiem Public Domain).

- **SnoopDos 1.7** - kolejna, ulepszona wersja niezwykle użytecznego programu autorstwa Eddy Carrolla. Dzięki prowadzonemu przez ten program nadzorowi wszelkich operacji dyskowych (via DOS) i wypisywaniu nazw plików (odczytywanych, zapisywanch lub kasowanych), będziesz mógł z łatwością odnaleźć przyczynę nieuruchamiania się twoich ulubionych programów (którą np: może okazać się tylko brak odpowiedniej biblioteki). Możliwość dowolnego skonfigurowania programu np: wyłączenie śledzenia funkcji Open() czy zapisywanie sesji pliku na dysku. Dostępna jest ponoć wersja 2.0, lecz jak dowiedzieliśmy się od samego autora, jest to przerabiana przez osoby trzecie wersja 1.4.

- **AGA IFF Converter 2.0** - nowy konwerter graficzny niezbędny dla każdego programisty. Możliwość konwersji plików IFF do formatu RAW NORMAL, RAW BLITTER, SPRITES, nagrywanie palety kolorów jako COPPER-LIST lub jako samą paletę, współpraca z każdym modelem Amigi (także wyposażonymi w OS 3.0). A także, jako jedyny na razie tego typu konwerter, umożliwia pracę z układami AGA (obsługa nowych trybów: 128/256 kolorów oraz HAM8!).

- **SysInfo 3.11** - kolejna edycja tego popularnego programu do testowania szybkości i konfiguracji komputera. Widoczne zmiany w stosunku do poprzednich wersji to np: wprowadzenie nowych modeli Amigi jako odnośnik przy testowaniu szybkości lub możliwość wykrycia nowych układów graficznych (AGA) z czym nie radziły sobie poprzednie wersje.

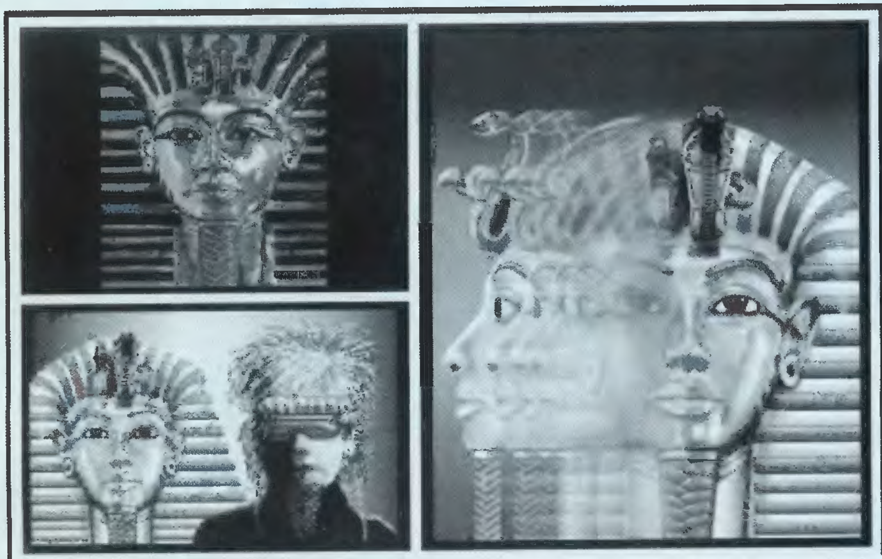
- **EditKeys 1.3** - program umożliwiający definiowanie własnej mapy klawiszy (keymap) czyli dowolne przypisanie dowolnego znaku (lub ciągu znaków) dowolnemu klawiszowi na klawiaturze Amigi. Program współpracuje ze wszystkimi modelami Amigi i pozbawiony jest wad rozpowszechnionego programu SetKey.

- **CPU Blit 1.0** - użytkownicy Amigi wyposażonej w procesor conajmniej 68020 mają możliwość, przy pomocy tego programu, podmienić w funkcjach bibliotek systemowych (głównie graficznej) operacje wykorzystujące blitter na analogiczne ale z użyciem procesora, co np: w przypadku scrolowania tekstu pozwala uzyskać dwukrotne przyspieszenie operacji. Duża ilość opcji pozwala dowolnie konfigurować ten program.

- **MemoryViewer 2.0** - krótki programik pozwalający przejrzeć całą dostępną pamięć typu Chip (której wielkość automatycznie rozpoznaje) jako dane graficzne. Zaletą programu jest pełna współpraca z systemem operacyjnym za pomocą input handlera, możliwość zmiany klawiszy aktywujących program oraz współpraca z AA ChipSet. (OS 2.0 lub wyższy!)

- **WBVerlauf** - następny krótki programik, tym razem przeznaczony dla użytkowników Amigi z AGA ChipSet i systemem operacyjnym 3.0, a pozwalający ustawić na ekranie barwną tęczę z wykorzystaniem pełnej 24-bitowej palety kolorów oferowanej przez nowe układy graficzne. Możliwość dowolnej zmiany zakresów kolorów, jej położenia jak również wykorzystanego koloru.

Dyskietka kosztuje 30 tys. złotych. Kwotę tę należy wpłacić na nasze konto, którego numer umieszczony jest na drugiej stronie okładki. Na odwrocie przekazu należy zaznaczyć czego wpłata dotyczy. Równocześnie można zamówić poprzednie numery (1 i 2) - każdy z nich kosztuje 25 tys. złotych. Termin realizacji zamówienia: do 4 tygodni.



Kupon ogłoszeniowy

imię i nazwisko

adres

treść:



Dr Boczek donosi...

W najbliższym czasie Sejm będzie rozpatrywał projekt ustawy o przestrzeganiu wartości w programach komputerowych. Odbędą się głosowania mające rozstrzygnąć o tym, że:

- jedynymi dopuszczalnymi będą wyłącznie wartości dodatnie (najbardziej znaczący bit będzie ustawowo ignorowany).

- zabrania się używania w programach opcji Abort. Sugerowanym słowem do nazywania tego typu procedur jest Amen.

- gry o bohaterach mitologicznych, lub też takie gdzie występują postacie typu diabły i duchy, będą zabronione. Niektóre kluby poselskie zapowiedziały, iż zgłoszą poprawkę, aby gry te były jednak dozwolone, ale od lat osiemnastu.

- zgodę na zainstalowanie modemu będzie wydawał sysop najbliższego parafialnego BBS'u, za zgodą proboszcza. Święcenia sprzętu nie będą jeszcze wymagane. Wkrótce jednak staną się one obowiązkowe i za poświęcenie Amigi opłata wynosić będzie według proponowanej taryfy tyle co za chrzest, a za poświęcenie peceta tyle co za pogrzeb.

- używanie wszelkich programów emulujących inny sprzęt będzie uznane za wysoce niemoralne i w konsekwencji zabronione.

- rozbieranie komputerów (w szczególności zdejmowanie obudowy) i oglądanie nagich chipów będzie dozwolone tylko fachowemu serwisowi legitymującemu się certyfikatem seminaryjnym.

- aby uniknąć niewłaściwych skojarzeń historycznych zabroni się odkładania rejestrów na stos. W bardziej liberalnej opcji, proponuje się ominięcie zakazu, poprzez zastąpienie słowa "stos" innym terminem np."kupka".

- usuwanie wirusów poczętych w pamięci komputera grozi karą do 10 lat pozbawienia Amigi. Zapobiegać wirusom można wyłącznie poprzez używanie 2 dysków (Workench i Extras) lub włączanie komputera w ustalonych dniach miesiąca, w czasie których aktywność wirusów jest znacznie mniejsza.

Stosowanie niedozwolonych środków (takich jak między innymi protectory na bootblock'ach, dokonujące krwawego Hard Resetu), będzie groziło karą pozbawienia Amigi do lat pięciu z możliwością zamiany kary na areszt. 1 dzień bez Amigi jest karą równą tygodniowemu aresztowi.

Czy wszystkie proponowane punkty ustawy zostaną wprowadzone w życie, czas pokaże...

Firma Intel konsekwentnie dementuje pogłoski, jakoby procesory serii 80x86, były we wstępnej fazie projektowane z myślą o zastosowaniu ich w programatorach pralek automatycznych, zamiast w komputerach. Argumenty Intela wydają się być nie do obalenia: niektóre pralki są zbyt skomplikowane, aby wcześniejsze procesory tej firmy zdołały sprostać zadaniu. Jako anegdotę można w tym miejscu dodać wypowiedź profesora mikrobiologii (mikrobiologia - nauka zajmująca się mikroprocesorami i biologią). Jeden ze studentów zadał mu takie pytanie:

- Jakie są cechy wspólne procesora Intela i dżdżownicy?

Na co profesor odpowiedział:

- Oprócz szybkości... jeszcze segmentacja.

Ostatnio pojawiły się na rynku bardzo tanie dyski no-name, których szczególną cechą było bardzo szybkie "padanie". Wynajęta przez jednego z poszkodowanych agencja detektywistyczna ustaliła, że dyski te pochodzą z Tajwanu, gdzie kilku oszustów kupuje tanie dyski 5,25" i osobno plastikowe, 3,5 calowe koperty. Następnie nożyczkami wycinają oni z dyskietek odpowiednie kształty i wkładają do obudów. Tak wyprodukowane no-name'y są bardzo tanie, lecz często ze względu na krzywe wycięcie, ulegają uszkodzeniu. Aby zabezpieczyć się przed oszustami, każdy nowo zakupiony dysk należy:

- otworzyć, wyjąć samą dyskietkę i dokładnie obejrzeć, czy aby dobrze wycięta. Jeżeli wszystko będzie w porządku, to jeżeli jeszcze się da, należy włożyć dysk z powrotem i zakleić.

- wodoodpornym pisakiem podpisać się na nośniku, aby później pożyczony dysk miał identyfikator właściciela. W ten sposób wykonane oznaczenie, będzie nie do usunięcia: wodą się nie zmyje, a wydrapanie zniszczyłoby nośnik. Takiego dysku na pewno nikt nie będzie próbował sobie przywłaszczyć.

Na podstawie swej bujnej wyobraźni:
Dr Boczek